TRUNCATED PLUG NOZZLE

COMPUTER PROGRAM

DOCUMENTATION

VOLUME II

(NASA-CR-132966) TRUNCATED PLUG NOZZLE COMPUTER PROGRAM DOCUMENTATION, VOLUME 2 (Notre Dame Univ.) 202 p

N73-72908

Unclas

00/99 05537

BY

DR. CHARLES R. HALL, JR.

Report UNDAS PD-601-2

JULY 1971

Prepared under Contract Nos.

NSR 15-004-029 and

NAS 8-25601

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MARSHALL SPACE FLIGHT CENTER

by

DEPARTMENT OF AEROSPACE AND MECHANICAL ENGINEERING
UNIVERSITY OF NOTRE DAME
NOTRE DAME, INDIANA

PROJECT DIRECTOR: DR. T. J. MUELLER

XIX. FUNCTION FBASE6

XIX. FUNCTION FBASE6

Function subprogram FBASE6 directs the solution of the plug nozzle base pressure. A method of characteristics solution is done from the non-characteristic line starting at the plug tip to the left-running characteristic intersecting the plug tip.

COMMON BLOCKS

The following COMMON blocks are used: AMB, BBBLK, BBLK, BLDM, BLK3A, CNRANG, CNTR, CORNER, CSBLK, DATBLK, D3BLK, ETABLK, F4BLK, GAS, PARAM, PTNOS, POLIP, SIGBLK, SIZE, SOLBLK, THETBK, TPN, and TRBBLK.

TPNZZL SUBROUTINES

Subroutines LINEAR and HYPER use FBASE6.

FBASE6 uses subroutines and functions: CALC, CPB, ERF, ERFD3, FLOW, FSTEP4, F2D, I1, I2, J1, J2, LIPSHK, LNEAR1, OSHOCK, PMANGL, RSBF, SHOCK, SIGMA, SOTE, SOTE2B, STRLNE, and TAB.

FORTRAN SYSTEM ROUTINES

Built-in FORTRAN functions ASIN, ATAN, SIN, SQRT, and TAN are used.

CALLING SEQUENCE

The calling sequence is:

X = FBASE6(PBP1,$,NEPS)

PBP1 is the base pressure ratio, P_b/P_1

$$\text{NEPS} = \begin{cases} 1 \text{ for axisymmetric flow} \\ 0 \text{ for planar flow} \end{cases}$$

AAST3A - isentropic area ratio just upstream of recompression

AB - base area (in^2)

ACST4 - accuracy requirement used in locating the j-streamline

B - term represented by Equation 2

CI2 - an evaluation of the I_2 integral (Equation 13)

CJ2 - an evaluation of the J_2 integral (Equation 15)

COTDEL - the cotangent of the change in streamline angle at a shock wave

CR - ratio of the Crocco number just before recompression to that just after separation

CS - square of the Crocco number just ahead of recompression

C1 - Crocco number just before separation

C2A - Crocco number just downstream of separation

C2AS - square of C2A

C3A - Crocco number just upstream of recompression

C3AC2A - equivalent to CR

C3AS - square of C3A

C3D - Crocco number at recompression on the d-streamline

C3DS - square of C3D

C5 - Crocco number on the d-streamline at recompression (Equation 16)

C6 - Crocco number on the d-streamline at recompression (Equation 9)

D - non-dimensional distance along the initial line

DEL - increment of distance along the starting line

DELMX - maximum increment of distance along the starting line

DELTA - change in streamline (radians) across a shock wave

DEN - denominator of a shock wave equation

DI - array of non-dimensional distances along the starting line

DJM1 - real number (= NDJ - 1)

DLINTL - the integral portion of Equation 20

DRS - density ratio across the shock

DSR - change in entropy across a shock divided by the gas constant

DTC - incremental change in streamline angle (radians) in going from the plug surface to the near wake

EGP - geometric parameter (left hand side of Equation 1)

EKJ - real number used to determine the J-subscript

ETAD3 - non-dimensional height of the d-streamline at recompression

ETAJ3 - non-dimensional height of the j-streamline at recompression

ETAM2D - non-dimensional difference between the viscous and inviscid coordinate systems just after separation

ETAM3D - non-dimensional difference between the viscous and inviscid coordinate systems at recompression

FP - pressure ratio function between separation and recompression

G - ratio of specific heats

GC - gravitational constant ($ft\text{-}lb_m/lb_f\text{-}sec^2$)

GD - mass bleed rate (lb_m/sec)

GP - geometric parameter (right hand side of Equation 1)

G1 - ratio of specific heats function

G2 - ratio of specific heats function

G3 - ratio of specific heats function

G4 - ratio of specific heats function

G5 - ratio of specific heats function

H - non-dimensional bleed number (Equation 19)

HTEST - non-dimensional bleed number (Equation 20)

I - subscript

IBOUND - integer telling which type of base pressure solution is desired

IEND - last value of the I-subscript

IM - subscript

IP - subscript

IRECP - integer denoting whether recompression has been reached

IST - starting value of the I-subscript

J - subscript

JCORN - number of J-subscripts needed to negotiate the expansion from the top contour to the free jet boundary

JP - subscript

K - subscript

KPA - term representing Equation 18

K1 - K-1

K2 - K-2

LL4 - integer used in call statement to LINEAR

LSHK - integer denoting whether a "lip shock" is present

MAMB - Mach number corresponding to ambient conditions

MBLD - integer denoting whether base bleed and/or a boundary layer is present

MESHPM - integer used in setting the minimum number of discrete turns from the plug surface to the near wake

MI - integer used in determining whether I is odd or even

M1A - Mach number just before separation

M1AS - square of M1A

M2A - Mach number just after separation

M2AS - square of M2A

M3A - Mach number just upstream of recompression

M3AM2A - ratio of M3A to M2A

M3AS - square of M3A

NASHF - recompression coefficient

NDI - maximum number of I-subscripts

NDIM1 - NDI-1

NDJ - maximum number of J-subscripts

NOIPTS - number of points on the starting line

NOPPTS - number of plug coordinate points

NTC - number of discrete turns from the plug surface to the near wake

NU - array of Prandtl-Meyer angles (radians) at each point in the characteristic matrix

NUAMB - Prandtl-Meyer angle (radians) corresponding to MEMB

NUI - array of Prandtl-Meyer angles (radians) at each point along the starting line

PA - ambient pressure (lb/in^2)

PHID3 - velocity ratio on the d-streamline at recompression

PHIJ3 - velocity ratio on the j-streamline at recompression

PI - a constant (=3.14159265)

PNASH - pressure ratio incorporating the recompression coefficient

POLP - stagnation pressure downstream of the "lip shock" (lb/in^2)

POPA - ambient pressure ratio (P_{o1}/P_{at})

POR - stagnation pressure ratio across a shock

PO1 - chamber stagnation pressure (lb/in^2)

PYPX - static pressure ratio across a shock

P1P3 - static pressure before separation divided by the static pressure just upstream of recompression

P2P3 - static pressure just after separation divided by the static pressure just upstream of recompression

P4P3 - static pressure ratio across the recompression shock

R - array of non-dimensional radial coordinates at each point in the characteristic matrix

REMDST - remaining non-dimensional distance along the starting line

RG - gas constant ($ft-lb_f/lb_m-^oR$)

RGG - the square root term of Equation 19

RI - array of non-dimensional radial coordinates at each point along the initial line

RMPTS - remaining number of points on the initial line

RP - array of non-dimensional radial coordinates at each point on the plug surface

RPB - plug base radius (inches)

RSRB - wake radius ratio

RTRIG - trigonometric portion of Equation 20

R2 - non-dimensional plug base radius

S - array of entropies at each point in the characteristic matrix ($ft-lb_f/lb_m-^oR$)

SA1R - shock wave angle of the "lip shock"

SI - array of entropies at each point along the initial line ($ft-lb_f/lb_m-^oR$)

SINSA - the sine of the shock wave angle

SOD - sum of the distances along the initial line

SOL3A - equivalent to M3AM2A, OR the streamline angle (radians) at recompression

S3A - jet spread parameter (Equation 7)

T - array of streamline angles (radians) at each point in the characteristics matrix

THET12 - change in streamline angle from the plug surface to the near wake

TI - array of streamline angles (radians) at each point in the characteristics matrix

TLEN - incremental streamline angle (radians) in going from the top contour to the free jet boundary

TOP - numerator of a shock wave equation

TO1 - chamber stagnation temperature (^{o}R)

TP - array of streamline angles (radians) at each point along the surface of the plug

TRB - base temperature ratio, T_b/T_{o1}

TRFD3 - temperature effects on the d-streamline at recompression

TRFJ3 - temperature effects on the j-streamline at recompression

TRS - static temperature ratio across a shock

TST - a test for convergence of LINEAR

T2 - streamline angle (radians) at the last point on the plug surface

VI1D - value of the I_1 integral (Equation 12)

VI1J - value of the I_1 integral (Equation 12)

VJ1D - value of the J_1 integral (Equation 14)

VJ1J - value of the J_1 integral (Equation 14)

VSQ - square of the shock angle necessary to give the desired base pressure ratio

W - shock wave angle (radians)

X - array of non-dimensional axial coordinates at each point in the characteristics matrix

SOLUTION METHOD

Skip a page and print out the estimate of the base pressure ratio.

1. PRINT 103

2. PRINT PBP1

Set lip shock parameters to zero.

3. IRECP = 0

4. LSHK = 0

See if a compression at the near wake is encountered.

5. If PBP1 > 1.0, then LSHK = 1

Define calculational quantities.

6. NDIM1 = NDI - 1

7. DJM1 = NOIPTS - 1

Calculate the ambient pressure ratio and the corresponding Mach number and Prandtl-Meyer (P-M) angle.

8. POPA = PO1/PA

9. MAMB = SQRT((2/(G-1))(POPA$^{\frac{G-1}{G}}$) - 1))

10. NUAMB = PMANGL(MAMB,G)

Set the characteristic matrix to zero.

11. DO 17 I = 1,NDI

12. DO 17 J = 1,NDJ

13. X(I,J) = 0

14. R(I,J) = 0

15. NU(I,J) = 0

16. T(I,J) = 0

17. S(I,J) = 0

Redefine the sum of the linear distances from point to point along the initial line.

18. SOD = DI(NOIPTS)

Set the first characteristic point.

19. X(1,1) = XI(1)

20. R(1,1) = RI(1)

21. NU(1,1) = NUI(1)

22. T(1,1) = TI(1)

23. S(1,1) = SI(1)

Calculate the number of vertical points needed to expand about the tip of the plug to the near wake.

24. JCORN = NTC + 1

Determine the remaining number of points.

25. RMPTS = NDJ - JCORN

Calculate the distance into which JCORN points will be placed.

26. TLEN = (JCORN - 1)(0.0025/M1A)

Determine the remaining distance.

27. REMDST = SOD - TLEN

Calculate the spacing increment in the larger region.

28. DELMX = REMDST/RMPTS

The term D is measured along the initial line.

29. D = 0.00

The subscript I is set.

30. I = 1

Set up the iniitial characteristic profile.

31. DO 39 J = 2,NDJ

Determine the spacing between points.

32. DEL = 0.00025/M1A

33. If J > JCORN, then DEL = DELMX

Increment distance.

34. D = D + DEL

Locate the initial line by linear interpolation.

35. X(I,J) = TAB(D,DI,XI,NOIPTS,1)

36. R(I,J) = TAB(D,DI,RI,NOIPTS,1)

37. NU(I,J) = TAB(D,DI,NUI,NOIPTS,1)

38. T(I,J) = TAB(D,DI,TI,NOIPTS,1)

39. S(I,J) = TAB(D,DI,SI,NOIPTS,1)

Now begin calculating along right-running characteristics starting at the initial data line. The solution continues to the left-running characteristic from the plug tip.

40. DO 55 K = 2,NDJ

Set the last subscript of I

41. IEND = K

The term EKJ is used to determine the J subscript.

42. EKJ = K + 0.60

Begin moving along the right-running characteristics.

43. DO 54 I = 2,IEND

Define a calculational integer.

44. IM = I - 1

Set the J subscript.

45. EKJ = EKJ - 0.50

46. J = EKJ

See if I is odd or even.

47. MI = MOD(I,2)

48. If MI = 0, then GO TO 52

The subscript I is odd. Define a J index.

49. JP = J + 1

Calculate the characteristics point.

50. CALL CALC

51. GO TO 54

The subscript I is even. Define another J index.

52. JP = J - 1

Calculate the characteristics point.

53. CALL CALC

54. CONTINUE

55. CONTINUE

Now the characteristics calculations will proceed along right-running characteristics which start on the external free jet (constant pressure) boundary. Again the solution continues to the left-running characteristic from the plug tip.

56. DO 77 K = 3,NDIM1,2

Set the last value of the subscript I.

57. IEND = IEND + 1

Define two calculational integers.

58. K2 = K - 2

59. K1 = K - 1

Set the J subscript on the boundary.

60. J = NDJ

Calculate the boundary point.

61. CALL CPB

Determine the starting value of I

62. IST = K + 1

See whether the last point has been reached.

63. If IST > NDIM1, then GO TO 77

The term EKJ is used to determine the J subscript.

64. EKJ = NDJ + 0.60

Begin moving along the right-running characteristic.

65. DO 76 I = IST,IEND

Define an I – integer.

66. IM = I - 1

Define the J subscript.

67. EKJ = EKJ - 0.50

68. J = EKJ

See if I is odd or even.

69. MI = MOD(I,2)

70. If MI = 0, then GO TO 74

The subscript I is odd. Define another J index.

71. JP = J + 1

Calculate the characteristic point.

72. CALL CALC

73. GO TO 76

The subscript I is even. Define another J index.

74. JP = J - 1

Calculate the characteristic point.

75. CALL CALC

76. CONTINUE

77. CONTINUE

Skip a page.

78. PRINT 103

Define five ratios of G.

79. G1 = (G - 1)/2

80. G2 = (G + 1)/2

81. G3 = (G - 1)/G

82. G4 = G/(G-1)

83. G5 = (G + 1)/((2)(G - 1))

Define the constant π and the gravitational constant.

84. PI = 3.14159265

85. GC = 32.18

Define a calculational ratio.

86. RGG = RG/((G)(GC))

Calculate the base area.

87. AB = (PI)(RPB^2)

88. If NEPS = 0, then AB = (2)(RPB)

Calculate the non-dimensional bleed number. (Equation 19)

89. H = ((GD)(SQRT(TO1))/((AB)(PO1)))(SQRT(RGG))

Invert the base temperature ratio.

90. TR = 1/TRB

Check for error condition.

91. If PBP1 < 0, then GO TO 195

Determine the square of the Mach number just ahead of separation and the Crocco number after the expansion.

92. M1AS = $M1A^2$

93. C1 = (1 + (G1)(M1AS))$(PBP1)^{-G3}$

Check for a compressive turn.

94. If PBP1 < 1.00, then GO TO 107

A compressive turn is indicated. First calculate the streamlines in the characteristic matrix.

95. CALL STRLNE(I)

A shock is present. Now determine what shock wave angle will give the desired base pressure ratio.

96. VSQ = ((PBP1)(G+1) + (G-1))/((2)(G)(M1AS)

Calculate the sine of the shock angle.

97. SINSA = SQRT(VSQ)

Determine the shock wave angle.

98. SA1R = ASIN(SINSA)

Now calculate the change in streamline angle for this compressive turn. First calculate the numerator of the expression.

99. TOP = (G+1)(M1AS)

And now the denominator.

100. DEN = (2)((M1AS)(VSQ) - 1)

Now calculate the change in streamline angle.

101. COTDEL = (TAN(SA1R))((TOP/DEN) -1)

102. DELTA = ATAN(1/COTDEL)

Calculate properties downstream of this shock.

103. CALL OSHOCK

Square the downstream Mach number and redefine the change in streamline angle.

104. M2AS = $M2A^2$

105. THET12 = - DELTA

106. GO TO 110

An expansion at the plug tip is present. Calculate the Mach number after the expansion.

107. M2AS = (C1 - 1)/G1

108. M2A = SQRT(M2AS)

Calculate the change in streamline angle through the expansion.

109. THET12 = PMANGL(M2A,G) = PMANGL(M1A,G)

Calculate the wake radius ratio.

110. RSRB = RSBF(M1A,G)

Print out the Mach number upstream of separation, the wake radius ratio, the Mach number immediately after separation and the change in streamline angle through separation.

111. PRINT M1A, RSRB

112. PRINT M2A, THET12

Calculate the increment of streamline angle change through the expansion.

113. DTC = - THET12/NTC

Calculate the Crocco number immediately after separation.

114. C2AS = M2AS/(M2AS + 1/G1)

115. C2A = SQRT(C2AS)

Calculate the flow field up to recompression. The term IBOUND = 1 for a conetail solution, while IBOUND = 2 for a constant pressure boundary near wake solution.

116. If IBOUND = 1, and LSHK = 1, then CALL LIPSHK(1,NEPS,$)

117. If IBOUND = 1, and LSHK = 0, then CALL FLOW (1,NEPS,$)

Determine the Mach number ratio.

118. If IBOUND = 1, then M3AM2A = SOL3A

119. If IBOUND = 2, then M3AM2A = 1.0

Calculate the Mach number at recompression and its square.

120. M3A = (M2A)(M3AM2A)

121. M3AS = M3A^2

Calculate the corresponding Crocco number squared.

122. C3AS = M3AS/(M3A + 1/G1)

Redefine the square of the Crocco number and calculate the Crocco number.

123. CS = C3AS

124. C3A = SQRT(C3AS)

Determine the ratio of Crocco number and redefine.

125. C3AC2A = C3A/C2A

126. CR = C3AC2A

Determine the jet spread parameter. (Equation 7)

127. S3A - SIGMA(M3A,G,TR,NEPS)

Calculate the static pressure ratio.

128. P2P3 = ((1+(G1)(M3AS))/(1+(G1)(M2AS)))^{G4}

See whether the flow is planar or axisymmetric.

129. If NEPS = 0, then GO TO 141

The flow is axisymmetric. Calculate the streamline angle of the near wake.

130. THT12 = THET12 - T2

Calculate the geometric parameter. (Equation 8)

131. GP = ((S3A)(TAN(THT12))/(1/RSRB-1))^2

Print the jet spread parameter and the geometric parameter.

132. PRINT S3A, GP

Calculate an auxiliary equation.

133. FP = (3)(G3)(1-P2P3)/((C2A()C3A))

Evaluate the J2 and I2 integrals.

134. CJ2 = J2(-3,+3)

135. C12 = I2(-3,+3)

Locate the displacement streamline just after separation.

136. ETAM2D = 3 - (1-C2AS)(CI2)

Change the accuracy requirement on the streamline location when base bleed is present.

137. If MBLD $\neq$ 0, then ACST4 = 0.000010

Print out variables.

138. PRINT C2A, M2A, C3A, M3A, THT12

Locate the j-streamline.

139. CALL SOTE2B(FSTEP4,-3,+3, ACST4, ETAJ3,+1,$143)

140. GO TO 144

The flow is planar. Locate the j-streamline.

141. CALL SOTE (F2D,-3,+3,ACST4, ETAJ3,$143)

142. GO TO 144

No solution was found for the j-streamline.

143. RETURN 2

Calculate the velocity ratio on the j-streamline, and print out the location and velocity ratio of this streamline.

144. PHIJ3 = (0.5)(1 + ERF(ETAJ3))

145. PRINT ETAJ3, PHIJ3

Determine non-isoenergetic mixing effects.

146. TRFJ3 = PHIJ3 +(TRB)(1-PHIJ3)

Calculate the Crocco number along the d-streamline. (Equation 16)

147. C5 = (PHIJ3)(C3A)/SQRT(TRFJ3)

Calculate the streamline angle at recompression for a conetail mixing solution.

148. If IBOUND = 1, then THET3A = THET12 - T2

Calculate the flow field for a constant pressure mixing solution.

149. If IBOUND = 2, and LSHK = 0, CALL FLOW(1,NEPS,$)

150. If IBOUND = 2, and LSHK = 1, CALL LIPSHK(1,NEPS,$)

Determine the streamline angle at recompression for a constant pressure mixing solution.

151. If IBOUND = 2, then THET3A = -SOL3A

Calculate the shock wave angle at recompression.

152. CALL SHOCK

Calculate the static pressure rise across this oblique recompression shock.

153. P4P3 = $((G)(M3AS)(SIN(W))^2 - G1)/G2$

Include the effects of a recompression coefficient.

154. PNASH = (NASHF)(P4P3) + (1-NASHF)(P2P3)

Calculate the Crocco number on the d-streamline from the pressure rise information. (Equation 9)

155. C6 = $SQRT(1 - 1/PNASH^{G3})$

See whether base bleed is present.

156. If MBLD = 0, then GO TO 188

Base bleed (and/or boundary layer) is present. Redefine the Crocco number on the d-streamline and its square.

157. C3D = C6

158. C3DS = $C3D^2$

Calculate the first estimate of the velocity ratio on this streamline.

159. PHID3 = C3D/C3A

The velocity ratio on the d-streamline will vary for non-isoenergetic mixing. An iteration scheme is used.

160. DO 162 IP = 1,50

Calculate base heating (or cooling) effects, and modify the velocity ratio. (Equation 16)

161. TRFD3 = PHID3 + (TRB)(1-PHID3)

162. PHID3 = C3D/((C3A)(SQRT(TRFD3))

Locate the d-streamline.

163. CALL LNEAR1

Print out variables.

164. PRINT C3D,PHIJ3,ETAJ3,PHID3,ETAD3

See if the flow is axisymmetric or planar.

165. If NEPS = 0, then GO TO 180

The flow is axisymmetric. Evaluate the J1 and I1 integrals up to the j- and d- streamlines. (Equations 12-15)

166. VJ1J = J1(-3,ETAJ3)

167. VI1J = I1(-3,ETAJ3)

168. VI1D = I1(3,ETAD3

169. VJ1D = J1(-3,ETAD3)

Print out the values of the integrals and the value of B.

170. PRINT VJ1J, VI1J, VI1D, VJ1D, B

Calculate the integral terms in the base bleed equation.

171. DLINTL = (VJ1D-VJ1D) - (B)(VI1D-VI1J)

Calculate the trigonemetric terms in the base bleed equation.

172. RTRIG = 1/((TAN(THET3A))(SIN(THET3A)))

Calculate the one-dimensional area ratio.

173. AAST3A = $(G2^{G5})(M3A)(1 + (G1)(M3AS))^{-G5}$

174. AAST3A = 1/AAST3A

Calculate an auxiliary expression. (Equation 18)

175. KPA = $(8)((2/(G+1))^{G5})(1\text{-}C3AS)/((AAST3A)(SQRT(RGG)))$

Print out results.

176. PRINT DLINTL, RTRIG, AAST3A,KPA

Calculate the non-dimensional bleed number. (Equation 20)

177. HTEST = (SQRT(RGG))(KPA)(RTRIG)(1-RSRB)^2(DLINTL)/((4)(S3A^2))

Determine the ratio of the two non-dimensional bleed numbers.

178. FBASE6 = H/HTEST

Print out the results.

179. PRINT FBASE6, H, HTEST

Return to calling program.

180. RETURN

The flow is planar. Evaluate the I1 integral to the j- and d-streamlines.

181. VI1J = I1(-3,ETAJ3)

182. VI1D = I1(-3,ETAD3)

Evaluate the integral term in the base bleed equation.

183. DLINTL = VI1D - VI1J

Modify the definition of the non-dimensional bleed number.

184. H = (H)(SQRT((GC)(G-1)/2))

Calculate the non-dimensional bleed number from the mixing solution.

185. HTEST = (C3A)(1-C3AS)^{G4}(DLINTL)/((S3A)(SIN(THET3A)))

Evaluate the ratio of these bleed numbers and print out the results, and then return to the calling program. (Equation 21)

186. FBASE6 = H/HTEST

187. PRINT FBASE6,H,HTEST

188. RETURN

No base bleed is present. The function is evaluated and is printed out. A return is then made to the calling program.

189. FBASE6 = C5-C6

190. PRINT FBASE6, C5, C6

191. RETURN

The detachment angle has been exceeded for the recompression shock. An attempt is made to rectify this situation. Calculate the static pressure ratio.

192. P1P3 = P2P3/PBP1

Redefine the pressure rise across the oblique shock and calculate the corresponding Crocco number on the d-streamline at recompression.

193. PNASH = (NASHF)(P1P3) + (1-NASHF)(P2P3)

194. C6 = SQRT(1 - 1/PNASH^{G3})

Repeat calculations.

195. GO TO 188

An error has developed in subroutine FLOW or LIPSHK, or PBP1 < 0.

196. RETURN 2

XX. SUBROUTINE SHOCK

XX. SUBROUTINE SHOCK

Subroutine SHOCK provides a closed-form solution to the shock wave cubic equation for an ideal gas. This subroutine returns the shock wave angle in radians given the initial Mach number and streamline angle in radians. If the detachment turn angle is exceeded, the value returned is the detachment turn angle.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Function subprogram and subroutines FBASE6, LIPSHK, and SSHAPE call SHOCK

SHOCK does not call any subroutines or functions.

FORTRAN SYSTEM ROUTINES

Built-in FORTRAN functions ACOS, ATAN, COS, SQRT, and TAN are used.

CALLING SEQUENCE

The calling sequence is:

CALL SHOCK (M1,DE,G,N,W,$)

M1 is the upstream Mach number.

DE change in streamline angle across the shock.

G is the ratio of specific heats.

$$N = \begin{cases} 4 \text{ for a weak shock solution} \\ 0 \text{ for a strong shock solution} \end{cases}$$

W is the shock wave angle measured from the upstream streamline angle.

A	- term in shock wave cubic equation solution
B	- term in shock wave cubic equation solution
C	- term in shock wave cubic equation solution
D	- the cotangent of the change in streamline angle through the shock
DD	- detachment turn angle (radians)
DM	- Mach number equation
G1	- term involving the ratio of specific heats
G2	- term involving the ratio of specific heats
M1S	- square of the upstream Mach number
P	- term in shock wave cubic equation solution
PHI	- angle in shock wave cubic equation solution (radians)
PHIARG	- cosine of PHI
P3	- a constant ($=\pi/3$)
TD	- tangent of the change in streamline angle through the shock
X	- square of the cotangent of the detachment turn angle
Y	- term used in calculating the shock wave angle

SOLUTION METHOD

Check for error conditions.

1. If M1 < 1, then GO TO 25

Define two ratios of G

2. G1 = (G - 1)/2

3. G2 = (G + 1)/2

Determine the cotangent of the streamline angle downstream of the shock.

4. TD = TAN(DE)

5. D = 1/TD

Define some addition auxiliary terms.

6. M1S = $M1^2$

7. DM = M1S - 1

Print out possible error conditions.

8. If D < 10^{-6}, or DM < 10^{-6}, then PRINT DM,DE

Define the coefficients.

9. A = 1 + (G1)(M1S)

10. B = -(DM)(D)

11. C = 1 - (G2)(M1S)

12. P = $(0.25/DM^3)((18)(A)(C)(DM) + (27)(A^2) - (DM^2)(C^2))$

Calculate the detachment turn angle.

13. X = $P/2 + SQRT((0.25)(P^2) + (A)(C/DM)^3)$

14. DD = ATAN(1/SQRT(X))

See if the given angle has exceeded the detachment turn angle.

15. If DE > DD, then GO TO 22

Calculate the shock wave angle, W.

16. PHIARG = ((9)(A)(B)(C) - (27)(A^2)(D) - (2)(B^3))/((2)(B^2 - (3)(A)(C))$^{3/2}$)

17. PHI = ACOS(PHIARG)/3

18. P3 = 1.04719755

19. Y = (2)(SQRT((B^2 - (3)(A)(C))(COS(PHI + (N)(P3)) - B

20. W = ATAN(Y/((3)(A))

21. RETURN

The detachment turn angle has been exceeded.

22. W = DD

23. PRINT 14

24. RETURN 6

The initial Mach number is less than unity.

25. PRINT 15

26. RETURN 6

XXI. SUBROUTINE OSHOCK

XXI. SUBROUTINE OSHOCK

Subroutine OSHOCK calculates the flow properties across an oblique shock wave. It returns the downstream Mach number, the static and total pressure ratios across the shock, the static temperature and density ratios across the shock, and the change in entropy across the shock.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Function subprogram and subroutines FBASE6, LIPSHK, and SSHAPE call OSHOCK.

OSHOCK does not call or use any subroutines or function subprograms.

FORTRAN SYSTEM ROUTINES

Built-in FORTRAN functions ALOG, SIN and SQRT are used.

CALLING SEQUENCE

The calling sequence is:

CALL OSHOCK (MX,W,DELTA,G,MY,PYPX,POYPOX,TYTX,RYRX,DSR)

MX is the upstream Mach number.

W is the shock wave angle.

DELTA is the change in streamline angle across the shock.

G is the ratio of specific heats.

MY is the downstream Mach number.

NOTE: In all of the following ratios, the downstream value is divided by the upstream value.

PYPX is the static pressure ratio.

POYPOX is the stagnation pressure ratio.

TYTX is the static temperature ratio.

RYRX is the density ratio

DSR is the change in entropy across the shock divided by the gas constant.

G1 - term involving the ratio of specific heats

G2 - term involving the ratio of specific heats

MN2 - square of the Mach number normal to the shock surface

Z - static pressure ratio across the shock

SOLUTION METHOD

Calculate the normal component of the Mach number downstream of the shock.

1. MN2 = ((MX)(SIN(W)))^2

Calculate the static pressure ratio across the shock.

2. PYPX = ((2)(G)(MN2) - (G-1))/G + 1)

Redefine the static pressure ratio.

3. Z = PYPX

Calculate the density ratio.

4. RYRX = ((G + 1)(Z) + (G + 1))/((G -1)(Z) (G + 1)

Determine the temperature ratio.

5. TYTX = Z/RYRX

Calculate the downstream Mach number.

6. MY = (SQRT(((G-1)(Z) + (G+1))/((2)(G)(Z))))/SIN(W - DELTA)

Define two ratios of G.

7. G1 = G/(G-1)

8. G2 = 1/(G-1)

Calculate the total pressure ratio across the shock.

9. POYPOX = (RYRX^{G1})/(Z^{G2})

Calculate the entropy change across the shock.

10. DSR = ALOG(1/POYPOX)

Check for possible error condition and rectify.

11. If MY > MX, and DSR > 0, then MY = (0.998)(MX)

12. RETURN

XXII. FUNCTION TAB

XXII. FUNCTION TAB

Subprogram TAB evaluates a single function for a given argument by interpolation between tabulated values. The maximum table size is 16384 entries, and the maximum degree of the interpolating polynomial is 10. Table search is by a binary method. For second order interpolation or higher, the interpolation is done by means of Lagrange polynomials.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Subroutines and functions FBASE6, LIPSHK, SETUP, SSHAPE, STRLNE, SURFP, and TPNZZL use TAB.

TAB does not call or use any subroutines or function subprograms.

FORTRAN SYSTEM ROUTINES

No built-in functions are used.

CALLING SEQUENCE

The calling sequence is:

Y = TAB(X,XT,YT,N,NORD)

X is the point on the ordinate at which the function is to be determined.

XT is an array of ordinate points.

YT is the corresponding array of abscissa points.

N is the number of points in the arrays.

NORD is the order of interpolation desired ($1 \leq NORD \leq 10$)

DI	- region in which a solution is to be found
DIL	- number of lower points to be used
DIU	- number of upper points to be used
GT	- "greater than" (used for output statement)
I	- subscript
IL	- lower number of points
IU	- upper number of points
I2	- regions of stored points
J	- subscript
L	- subscript
LT	- "less than" (used for output statement)
P	- LaGrange coefficient

SOLUTION METHOD

Check for error conditions.

1. If NORD > 10, then GO TO 53

2. If N > 12(15), then GO TO 27

3. If X < XT(1), then GO TO 29

4. If X > XT(N), then GO TO 31

Set increment.

5. DI = 0

Begin a binary search of the table argument.

6. DO 8 J = 1,14

7. If I2(J+1) ≥ N, then GO TO 10

8. CONTINUE

Solution was not found; too many points.

9. GO TO 27

Set the subscripts

10. I = I2(J)

11. I = I + DI

12. J = J - 1

Check for possible errors.

13. If J = 0, then GO TO 21

14. If I < N, then GO TO 17

Determine the I increment.

15. DI = -I2(J)

Restart cycle.

16. GO TO 11

Set final increment of I.

17. DI = I2(J)

18. If XT(I) > X, then DI = -DI

Make sure a solution does not occur exactly on a point.

19. If XT(I) ≠ X, then GO TO 11

20. If XT(I) = X, then GO TO 25

Check lower value of I

21. If XT(I) < X, then I = I + 1

Check order of interpolation.

22. If NORD > 1, then GO TO 33

A linear interpolation is indicated.

23. TAB = YT(1) - (YT(I) - YT(I-1))(XT(I) - X)/(XT(I) - XT(I-1))

Return to calling program.

24. RETURN

The solution falls exactly on a tabulated point.

25. TAB = YT(I)

26. RETURN

There are too many entries in the table.

27. PRINT N,I2(15)

Terminate program.

28. RETURN 0

The argument is below the table limit; return the lowest value in the table.

29. TAB = YT(1)

30. RETURN

The argument is above the table limit; return the highest value in the table.

31. TAB = YT(N)

32. RETURN

This next section deals with an order 2 through 10 interpolation. First determine NORD + 1 points to be used which are centered on X. An adjustment is made if X is near the end of the table.

First, set the upper and lower range of the subscript, I.

33. DIL = (NORD + 1)/2

34. DIU = NORD - DIL

Check for conflicts.

35. If I $\leq$ DIL, then I = DIL + 1

36. If I + DIU > N, then I = N - DIU

Set upper and lower point subscripts.

37. IL = I - DIL

38. IU = I + DIU

Calculate the Lagrange coefficients, P(L)

39. L = 0

40. DO 46 I = IL, IU

Increment L

41. L = L + 1

42. P(L) = 1

43. DO 46 J = IL, IU

44. If I = J, then GO TO 46

Calculate coefficients.

45. P(L) = (P(L))(X-XT(J))/(XT(I) - XT(J))

46. CONTINUE

Now evaluate the Lagrange polynomial at X.

First, set the subscript and function to zero.

```
47.  L = 0

48.  TAB = 0

49.  DO 51 I = IL,IU

50.  L = L + 1

51.  TAB = TAB + (P(L))(YT(I))
```

Return to calling program.

```
52.  RETURN
```

The order of interpolation is too great.

```
53.  PRINT NORD
```

Terminate program

```
54.  RETURN 0
```

XXIII. FUNCTION SIMR

XXIII. FUNCTION SIMR

Function subprogram SIMR performs a Simpson's Rule integration of a function between specified limits. The number of increments must be even.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

The following subroutine and function subprograms use SIMR; BLAYER, I1, I2, J1, and J2.

SIMR does not call any subroutines or function subprograms.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

```
A = SIMR(FCN,X1,XN,N)
```

FCN is the function to be integrated.

X1 is the lower limit of integration.

XN is the upper limit of integration.

N is the number of increments (N must be even).

DX - incremental spacing in the region

I - a counter indicating the number of separate calculations

SEV - sum of the even terms

SOD - sum of the odd terms

X - argument for the function

SOLUTION METHOD

The terms SEV and SOD are the sums of the even and odd terms respectively. Set these initially to zero.

1. SEV = 0

2. SOD = 0

Determine the increment used in the integration

3. DX = (XN-X1)/N

Begin calculating the odd or even terms at each increment.

4. DO 8 I = 2, N, 2

Locate the point.

5. X = X1 + (I)(DX)

Determine the sum of the odd terms.

6. SOD = SOD + FCN(X)

Reset X for an even term and then evaluate the sum of the even terms.

7. X = X - DX

8. SEV = SEV + FCN(X)

Reset sums of odd or even terms.

9. SOD = (2)(SOD)

10. SEV = (4)(SEV)

Evaluate the integration.

11. SIMR = (FCN(X1) + SOD + SEV - FCN(XN))(DX/3)

Return to calling program.

12. RETURN

XXIV. FUNCTION PMANGL

XXIV. FUNCTION PMANGL

Function subprogram PMANGL evaluates the Prandtl-Meyer (P-M) turn angle when given the Mach number and ratio of specific heats.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

The following programs use PMANGL: FBASE6, FLOW, LIPSHK, MAIN, SSHAPE, STRLNE, and TPNZZL.

PMANGL does not call any programs.

FORTRAN SYSTEM ROUTINES

Built-in FORTRAN functions ATAN and SQRT are used.

CALLING SEQUENCE

The calling sequence is:

NU = PMANGL(M,G)

M is the Mach number.

G is the ratio of specific heats.

A - term involving the ratio of specific heats

BETA - term involving the Mach number

SOLUTION METHOD

Evaluate a ratio G.

1. A = SQRT((G-1)/(G+1))

Determine a Mach number expression.

2. BETA = SQRT(M^2 - 1)

Calculate the Prandtl-Meyer angle in radians and return to calling program.

3. PMANGL = ATAN((A)(BETA))/A - ATAN(BETA)

4. RETURN

XXV. FUNCTION SIGMA

XXV. FUNCTION SIGMA

Function subprogram SIGMA evaluates the jet spread parameter which is required in a base pressure solution. For axisymmetric configurations, the jet spread parameter developed by Channapragada is used, whereas for planar geometries, Korst's value of the jet spread parameter is employed.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Function subprogram FBASE6 calls SIGMA

SIGMA does not call any programs.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

S3A = SIGMA(M,G,TR,NEPS)

M is the Mach number at recompression.

G is the ratio of specific heats.

TR is the temperature ratio, T_{o_1}/T_b.

$$NEPS = \begin{cases} 1 \text{ for axisymmetric flow} \\ 0 \text{ for planar flow} \end{cases}$$

CS - square of the Crocco number just upstream of recompression

R - empirical compressible divergence factor

SR - multiplication factor

SOLUTION METHOD

See if the flow is axisymmetric or planar.

1. If NEPS = 0, then GO TO 10

The flow is axisymmetric. Evaluate the square of the Crocco number.

2. $CS = M^2/(2/(G-1) + M^2)$

Two matching solutions are used depending on the value of CS.

3. If $CS \leq 0.70$, then GO TO 6

The compressible divergence factor is defined.

4. R = 0.25

Continue calculations.

5. GO TO 7

Set the compressible divergence factor.

6. $R = 0.5 - (0.715)(CS) + (0.4945)(CS^{1.9})$

Calculate the ratio of jet spread parameter to its incompressible value. (Equation 7)

7. $SR = 1/((R)(1 + (TR)(1 - CS)))$

Calculate the actual value of the jet spread parameter and return.

8. SIGMA = (SR)(12)

9. RETURN

The flow is planar. Calculate the jet spread parameter directly. (Equation 24)

10. SIGMA = 12.0 + (2.758)(M)

11. RETURN

XXVI. FUNCTION RSBF

XXVI. FUNCTION RSBF

Function subprogram RSBF determines the wake radius to base radius ratio at which recompression is assumed to occur. This is an analytic curve fitted to the data of Chapman.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Function subprogram FBASE6 uses RSBF.

RSBF does not use any other programs.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

RSRB = RSBF(M1,G)

M1 is Mach number before separation.

G is the ratio of specific heats.

SOLUTION METHOD

Evaluate function and return.

1. RSBF = 0.302 + 0.462/M1

2. RETURN

XXVII. FUNCTION ERF

XXVII. FUNCTION ERF

Function subprogram ERF evaluates the error function for a given argument.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

ERF is used by function subprograms: AI1, AI2, AJ1, AJ2, ERFD3, and FBASE6.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

```
Y = ERF(X)
```

X is the argument for which the error function will be evaluated.

A1 - coefficient of polynomial

A2 - coefficient of polynomial

A3 - coefficient of polynomial

A4 - coefficient of polynomial

A5 - coefficient of polynomial

DENOM - denominator of the equation

Y - absolute value of the argument

SOLUTION METHOD

Redefine the argument.

1. Y = ABS(X)

Specify coefficients.

2. A1 = 0.14112821

3. A2 = 0.08864027

4. A3 = 0.02743349

5. A4 = -0.00039446

6. A5 = 0.00328975

Calculate the term in the denominator.

7. $\text{DENOM} = (1 + (A1)(Y) + (A2)(Y^2) + (A3)(Y^3) + (A4)(Y^4) + (A5)(Y^5))^8$

Evaluate the error function.

8. ERF = 1 - 1/DENOM

The error function is an odd function.

9. If X < 0, then ERF = -ERF

Return to calling program.

10. RETURN

XXVIII. FUNCTION FSTEP4

XXVIII. FUNCTION FSTEP4

Function subprogram FSTEP4 performs calculations required in the base pressure solution. This function is used as an argument in the subroutine SOTE2B and is used to converge on the j-streamline.

COMMON BLOCKS

COMMON blocks BBLK, CSBLK, and F4BLK are used.

TPNZZL SUBROUTINES

FBASE6 uses FSTEP4

FSTEP4 uses function subprograms I1 and J2.

FORTRAN SYSTEM ROUTINES

Built-in FORTRAN function SQRT is used.

CALLING SEQUENCE

The calling sequence is:

```
Y = FSTEP4(ETAJ)
```

ETAJ is the non-dimensional location of the j-streamline.

B - result of Equation 2

CI2 - value of the I_2 integral (Equation 13)

CJ2 - value of the J_2 integral (Equation 15)

CR - ratio of the Crocco number at recompression to that immediately downstream of separation

CS - square of the Crocco number just upstream of recompression

EGP - geometric parameter (left hand side of Equation 1)

ETAM3D - non-dimensional coordinate shift between the viscous and inviscid coordinate systems at recompression

FP - pressure function

GP - geometric parameter (right hand side of Equation 1)

VI1 - value of the I_1 integral (Equation 12)

VJ1 - value of the J_1 integral (Equation 14)

SOLUTION METHOD

Calculate the J1 and I1 integrals.

1. VJ1 = J1(ETAJ,+3)

2. VI1 = I1(ETAJ,+3)

Determine the value of B (Equation 2)

3. B = ((CR)(CJ2) - VJ1)/((CR)(CI2) - VI1 + FP)

Calculate the geometric parameter. (Equation 1)

4. EGP = $(B-3)^2$ + (2)(B)(1-CS)(VJ1) - (2)(1-CS)(VJ1)

Determine the difference between the geometric parameter calculated above and that obtained from the geometry of the near wake. The solution for ETAJ is obtained when these two values agree.

5. FSTEP4 - EGP - GP

Locate the d-streamline. (Equation II-29)

6. ETAM = B + SQRT(GP)

7. RETURN

XXIX. FUNCTION J2

XXIX. FUNCTION J2

Function subprogram J2 evaluates the J2 integral used in the base pressure solution.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

J2 is used by function subprogram FBASE6.

J2 uses function subprograms AJ2 and SIMR

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

VJ2 = J2(ETAJ, + 3)

ETAJ is the non-dimensional location of the j-streamline

SOLUTION METHOD

Determine the J2 integral.

1. J2 = SIMR(AJ2, ETA1, ETA2, 30)

2. RETURN

XXX. FUNCTION J1

XXX. FUNCTION J1

Function subprogram J1 evaluates the J1 integral used in the base pressure solution.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Function subprograms FBASE6 and FSTEP4 use J1

J1 uses function subprograms AJ1 and SIMR

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

VJ1 = J1(ETAJ, +3)

ETAJ is the non-dimensional location of the j-streamline.

SOLUTION METHOD

Evaluate J1 integral and return.

1. J1 = SIMR(AJ1, ETA1, ETA2,30)

2. RETURN

XXXI. FUNCTION I2

XXXI. FUNCTION I2

Function subprogram I2 evaluates the I2 integral used in the base pressure solution.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Function I2 is used by function subprograms: FBASE6 and F2D.

I2 uses function subprograms AI2 and SIMR

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

VI2 = I2(ETA1, ETA2)

ETA1 is the lower limit of the non-dimensional streamline location.

ETA2 is the upper limit of the non-dimensional streamline location.

SOLUTION METHOD

Evaluate I2 integral and return to calling program.

1. I2 = SIMR(AI2, ETA1, ETA2,30)

2. RETURN

XXXII. FUNCTION I1

XXXII. FUNCTION I1

Function subprogram I1 evaluates the I1 integral which is used in the base pressure solution.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

I1 is used by the function subprograms FBASE6, FSTEP4, and F2D.

I1 uses function subprograms AI1 and SIMR.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

VI1 = I1(ETA1, ETA2)

ETA1 is the lower limit of the non-dimensional streamline location.

ETA2 is the upper limit of the non-dimensional streamline location.

SOLUTION METHOD

Evaluate the I1 integral and return to the calling program.

1. I1 = SIMR(AI1, ETA1, ETA2,30)

2. RETURN

XXXIII. FUNCTION AJ2

XXXIII. FUNCTION AJ2

Function subprogram AJ2 evaluates the integrand of the J2 integral at each point in the integration.

COMMON BLOCKS

COMMON blocks CSBLK and TRBBLK are used.

TPNZZL SUBROUTINES

Function subprogram SIMR uses AJ2

AJ2 uses function subprogram ERF

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

Y = AJ2(X)

X is the non-dimensional location of a streamline

A - velocity ratio along a streamline

AS - square of the velocity ratio along a streamline

CS - square of the Crocco number at recompression

TRB - base temperature ratio, T_b/T_{o_1}

TRF - portion of the demoninators of Equations 12-15

SOLUTION METHOD

Evaluate the velocity ratio.

1. A = (0.5)(1 + ERF(X)).

Calculate base heating effects, and square velocity ratio.

2. TRF = A + (TRB)(1-A)

3. AS = A^2

Evaluate the integrand at X and return to calling program. (Equation 15)

4. AJ2 = (AS)(X)/(TRF-(CS)(AS)).

5. RETURN

XXXIV. FUNCTION AJ1

XXXIV. FUNCTION AJ1

Function subprogram AJ1 evaluates the integrand of the J1 integral at each point in the integration.

COMMON BLOCKS

COMMON blocks CSBLK and TRBBLK are used.

TPNZZL SUBROUTINES

Function subprogram SIMR uses AJ1.

AJ1 uses function subprogram ERF.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

```
Y = AJ1(X)
```

X is the non-dimensional location of a streamline.

A - velocity ratio along a streamline

CS - square of the Crocco number at recompression

TRB - base temperature ratio, T_b/T_{o1}

TRF - portion of the denominators of Equations 12 - 15

SOLUTION METHOD

Evaluate the velocity ratio.

1. A = (0.50)(1 + ERF(X))

Determine base heating effects, and square the velocity ratio.

2. TRF = A + (TRB)(1-A)

3. AS = A^2

Evaluate the integrand at X and return to the calling program. (Equation 14)

4. AJ1 = (A)(X)/(TRF - (CS)(A^2))

5. RETURN

XXXV. FUNCTION AI2

XXXV. FUNCTION AI2

Function subprogram AI2 evaluates the integrand of I2 integral at each point in the integration.

COMMON BLOCKS

COMMON blocks CSBLK and TRBBLK are used.

TPNZZL SUBROUTINES

Function subprogram SIMR uses AI2

AI2 uses function subprogram ERF

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

Y = AI2(X)

X is the non-dimensional location of a streamline

A - velocity ratio along a streamline

AS - square of the velocity ratio along a streamline

CS - square of the Crocco number at recompression

TRB - base temperature ratio, T_b/T_{o_1}

TRF - portion of the denominators of Equations 12 - 15

SOLUTION METHOD

Evaluate the velocity ratio.

1. A = (0.50)(1 + ERF(X))

Determine base heating effects, and square the velocity ratio.

2. TRF = A + (TRB)(1-A)

3. AS = A^2

Evaluate the integrand at X and return to the calling program. (Equation 13)

4. AI2 = AS/(TRF - (CS)(AS))

5. RETURN

XXXVI. FUNCTION AI1

XXXVI. FUNCTION AI1

Function subprogram AI1 evaluates the integrand of the I1 integral at each point in the integration.

COMMON BLOCKS

COMMON blocks CSBLK and TRBBLK are used.

TPNZZL SUBROUTINES

Function subprogram SIMR uses AI1.

AI1 uses function subprogram ERF.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

```
Y = AI1(X)
```

X is the non-dimensional location of a streamline.

A - velocity ratio along a streamline

CS - square of the Crocco number at recompression

TRB - base temperature ratio, T_b/T_{o_1}

TRF - portion of the denominators of Equations 12 - 15

SOLUTION METHOD

Evaluate the velocity ratio.

1. A = (0.50)(1 + ERF(X))

Determine base heating effects.

2. TRF = A + (TRB)(1-A)

Evaluate the integrand at X and return to the calling program. (Equation 12)

3. AI1 = A/(TRF - (CS)(A^2))

4. RETURN

XXXVII. FUNCTION ERFD3

XXXVII. FUNCTION ERFD3

Function subprogram ERFD3 is used in conjunction with the subroutine SOTE to locate the d-streamline when given an error function velocity profile.

COMMON BLOCKS

COMMON block D3BLK is used.

TPNZZL SUBROUTINES

ERFD3 is used by the subroutine LNEAR1

ERFD3 uses the function subprogram ERF

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

Y = ERFD3(ETAD3)

ETAD3 is the non-dimensional location of the d-streamline.

PHI - calculated velocity ratio on the d-streamline

PHID3 - given velocity ratio on the d-streamline

SOLUTION METHOD

Calculate the velocity profile.

1. PHI = (0.50)(1 + ERF(ETAD3))

Obtain the difference between the velocity profile calculated above and that obtained in another program.

2. ERFD3 = PHI - PHID3

Return to calling program.

3. RETURN

XXXVIII. FUNCTION F2D

XXXVIII. FUNCTION F2D

Function subprogram F2D is used to evaluate the difference between I1 and I2 integrals in the planar base pressure solution.

COMMON BLOCKS

No COMMON blocks are used.

TPNZZL SUBROUTINES

Subroutine SOTE calls F2D

F2D uses function subprograms I1 and I2.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

Y = F2D(ETAJ)

ETAJ is the non-dimensional location of the J-streamline

VI1 - value of the I_1 integral (Equation 12)

VI2 - value of the I_2 integral (Equation 13)

SOLUTION METHOD

Evaluate the I1 and I2 integrals.

1. VI1 = I1(ETAJ, + 3)

2. VI2 = I2(-3,+3)

Subtract results of integrals. A solution is obtained when VI1 = VI2.

3. F2D = VI1 - VI2

Return to calling program.

4. RETURN

XXXIX. FUNCTION FMACH

XXXIX. FUNCTION FMACH

Function subprogram FMACH evaluates the coefficient of the mass flow function which is used to determine the amount of equivalent bleed due to the boundary layer.

COMMON BLOCKS

The COMMON block GAS is used.

TPNZZL SUBROUTINES

The subroutine BLAYER uses FMACH.

FMACH does not use any other programs.

FORTRAN SYSTEM ROUTINES

The built-in FORTRAN function SQRT is used.

CALLING SEQUENCE

The calling sequence is:

FMCH = FMACH(M)

M is the Mach number

C1 - part of the numerator of the mass flow equation

G - ratio of specific heats

GC - gravitational constant ($ft\text{-}lb_m/lb_f\text{-}sec^2$)

PO1 - chamber stagnation pressure (lb/in^2)

RG - gas constant ($ft\text{-}lb_f/lb_m\text{-}{}^oR$)

TO1 - chamber stagnation temperature (^oR)

SOLUTION METHOD

Define the gravitational constant.

1. GC = 32.174

Make and auxilliary calculation.

2. C1 = 1.0 + ((G-1)/2)(M^2).

Evaluate the function and return to the calling program.

3. FMACH = (M)(SQRT((G)(GC)(C1)/RG))

4. RETURN

XL. FUNCTION PRFLE

XL. FUNCTION PRFLE

Function subprogram PRFLE evaluates the integrand at each point of the integral used to determine the momentum thickness of the boundary layer.

COMMON BLOCKS

The COMMON block BLBK is used.

TPNZZL SUBROUTINES

The function subprogram SIMR uses PRFLE.

PRFLE does not use any other programs.

FORTRAN SYSTEM ROUTINES

No built-in FORTRAN functions are used.

CALLING SEQUENCE

The calling sequence is:

X = PRFLE(Y)

Y is the height in the boundary layer (inches)

C2AS - square of the Crocco number just after separation

DELTAI - boundary layer thickness (inches)

DEN - denominator of the profile equation

DL2ST - boundary layer momentum thickness (inches)

EX - boundary layer profile exponent

GDBL - equivalent mass bleed due to the boundary layer (lb_m/sec)

LAM - plug temperature ratio, T_p/T_{o_1}

N - denominator of the velocity profile exponent

NUM - numerator of the profile equation

VEL - velocity ratio along a streamline in the boundary layer

VEL2 - square of the velocity ratio along a streamline

YOD - non-dimensional height in the boundary layer

SOLUTION METHOD

Locate the height in the boundary layer.

1. YOD = Y/DELTAI

Evaluate the velocity ratio and its square. (Equation 23)

2. VEL = YOD^{EX}

3. VEL2 = $YOD^{(2)(EX)}$

Calculate the numerator and denominator of the integrand.

4. NUM = (VEL)(1 - VEL)

5. DEN = VEL + (LAM)(1-VEL) - (C2AS)(VEL2)

Evaluate the integrand, and return.

6. PRFLE = (1 - C2AS)(NUM/DEN)

7. RETURN

XLI. SUBROUTINE SEARCH

XLI. SUBROUTINE SEARCH

Subroutine SEARCH searches through the characteristics matrix to locate a point within the matrix and then returns the characteristic points which surround this known point. The solution involves comparing the y-intercepts of the characteristics with the y-intercept of the point at the same angle as the characteristic wave. Referring to Fig. XLI-1 below, an example of the calculation procedure is shown.

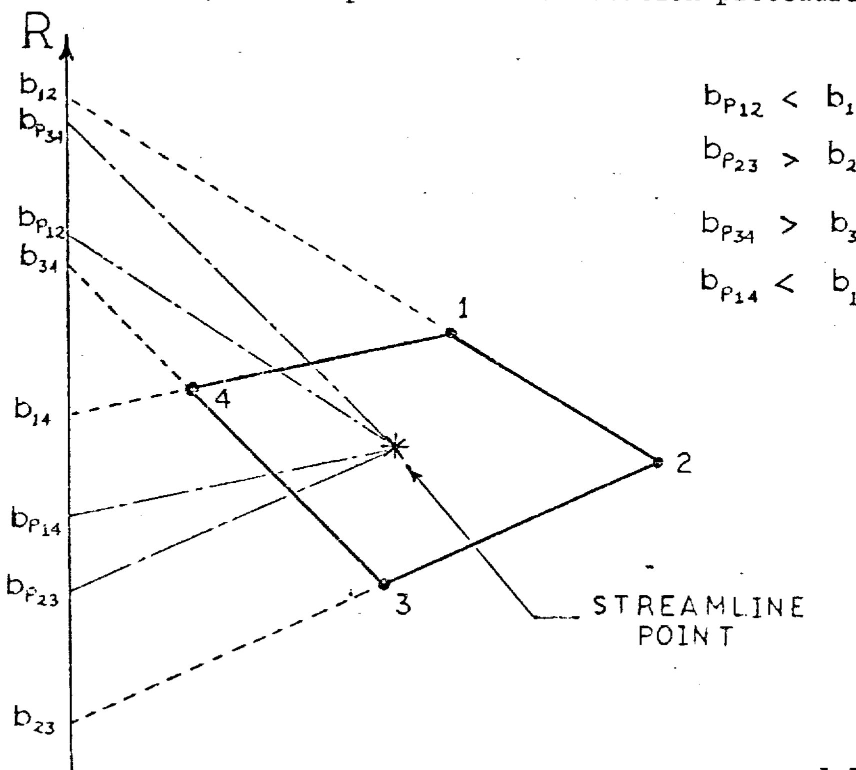


FIGURE XLI - 1

1. The y-intercept of the characteristic from point #1 to point #2, b_{12} is determined. The y-intercept of the given point, b_{P12}, is also calculated using the same slope as the characteristic. If $b_{P12} < b_{12}$, a solution is possible.

2. The y-intercept of the characteristic from point #2 to point #3, b_{23}, is determined. The y-intercept of the given point, b_{P23}, is also calculated using the same slope as that characteristic. If $b_{P23} > b_{23}$, a solution is possible.

3. The y-intercept of the characteristic from point #3 to point #4 is calculated, b_{34}. The y-intercept of the given point b_{P34}, is also calculated using the same slope as that characteristic. If

$b_{P34} > b_{34}$, a solution is possible.

4. The y-intercept of the characteristic from point #4 to point #1, b_{41}, is determined. The y-intercept of the given point, b_{P41}, is also calculated using the same slope as the characteristic. If $b_{P41} < b_{41}$, then a solution has been obtained.

Note that <u>all</u> of the conditions described in steps 1-4 must be met for a solution. The case shown here is one in which the characteristic points are in their "ideal" position. This program, however, treats not only this situation, but other orientations of the characteristic points. Each orientation must be considered separately. Diagrams are included.

COMMON BLOCKS

COMMON blocks PARAM, PLCBLK, and SIZE are used.

FORTRAN SYSTEM ROUTINES

The built-in function MOD is used.

CALLING SEQUENCE

The calling sequence is:

```
CALL SEARCH (X1,R1,NU1,T1,X2,R2,NU2,T2,X3,R3,NU3,T3,X4,R4,NU4,
   T4,XS,RS,A1,A2,A3,A4,NDF,$,IEND)
```

XI is the non-dimensional axial coordinate at point #1.

R1 is the non-dimensional radial coordinate at point #1.

NU1 is the Prandtl-Meyer angle (radians) at point #1.

T1 is the streamline angle (radians) at point #1.

X2 is the non-dimensional axial coordinate at point #2.

R2 is the non-dimensional radial coordinate at point #2.

NU2 is the Prandtl-Meyer angle (radians) at point #2

T2 is the streamline angle (radians) at point #2

X3 is the non-dimensional axial coordinate at point #3

R3 is the non-dimensional radial coordinate at point #3

NU3 is the Prandtl-Meyer angle (radians) at point #3.

T3 is the streamline angle (radians) at point #3.

X4 is the non-dimensional axial coordinate at point #4.

R4 is the non-dimensional radial coordinate at point #4.

NU4 is the Prandtl-Meyer angle (radians) at point #4.

T4 is the streamline angle (radians) at point #4.

XS is the non-dimensional axial coordinate of the given point.

RS is the non-dimensional radial coordinate of the given point.

A1 is a number indicating a top surface.

A2 is a number indicating a right surface.

A3 is a number indicating a bottom surface.

A4 is a number indicating a left surface.

NDF is the final I-subscript of the characteristics matrix.

IEND is an integer denoting that a solution was not found.

BP - y-intercept of the given point

B12 - y-intercept of the characteristic between points #1 and #2

B13 - y-intercept of the line between points #1 and #3

B23 - y-intercept of the characteristic between points #2 and #3

B24 - y-intercept of the line between points #2 and #4

B31 - equivalent to B13

B34 - y-intercept of the characteristic between points #3 and #4

B41 - y-intercept of the characteristic between points #4 and #1

B42 - equivalent to B24

I - subscript

ILOC - integer denoting what portion of the flow field is being calculated (See Fig. IV-1)

IPLC - integer denoting that the end of the characteristics matrix has been reached

ISTT - starting value of the I-subscript

J - subscript

JSTRT - starting value of the J-subscript

LOC - integer denoting which portion of the flow field is being calculated (See Fig. I-3)

MI - integer used in determining whether I is odd or even

NDFM1 - the final value of the I-subscript - 1

NDI - maximum number of I-subscripts in the characteristics matrix

NDJ - maximum number of J-subscripts in the characteristics matrix

NU - array of Prandtl-Meyer angles (radians) at each point in the characteristics matrix

R - array of non-dimensional radial coordinates at each point in the characteristics matrix

S - array of entropies at each point in the characteristics matrix ($ft\text{-}lb_f/lb_m\text{-}^{\circ}R$)

SL12 - slope of the characteristic between points #1 and #2

SL13 - slope of the line between points #1 and #3

SL23 - slope of the characteristic between points #2 and #3

SL24 - slope of the line between points #2 and #4

SL31 - equivalent to SL13

SL34 - slope of the characteristic between points #3 and #4

SL41 - slope of the characteristic between points #4 and #1

SL42 - equivalent to SL24

T - array of streamline angles (radians) at each point in the characteristics matrix

X - array of non-dimensional axial coordinates at each point in the characteristics matrix

SOLUTION METHOD

Define a calculational integer

1. NDFM1 = NDF-1

Assume for the present that the program will start with a point which will have four characteristic points surrounding it.

2. IPLC = 0

3. A1 = 1.0

4. A2 = 1.0

5. A3 = 1.0

6. A4 = 1.0

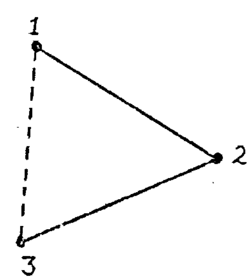
Check to see if the characteristics solution is in the throat region.

7. If LOC = 0, then GO TO 52

The characteristics matrix is assumed to be completely filled. Begin searching the left-hand side of the matrix. Point #4 does not exist for this case. Redefine the points. Move up the left side of the matrix.

8. DO 51 J = 2,NDJ

9. X1 = X(1,J)



10. R1 = R(1,J)

11. NU1 = NU(1,J)

12. T1 = T(1,J)

13. X2 = X(2,J)

14. R2 = R(2,J)

15. NU2 = NU(2,J)

16. T2 = T(2,J)

17. X3 = X(1,J-1)

18. R3 = R(1,J-1)

19. NU3 = NU(1,J-1)

20. T3 = T(1,J-1)

21. X4 = 0.0

22. R4 = 0.0

23. NU4 = 0.0

24. T4 = 0.0

Before checking the y-intercepts, see if the point has any possibility of being in this array of three points. If not, try another set of three points.

25. If XS > X1 and XS > X2 and XS > X3, then GO TO 51

26. If XS < X1 and XS < X2 and XS < X3, then GO TO 51

27. If RS > R1 and RS > R2 and RS > R3, then GO TO 51

28. If RS < R1 and RS < R2 and RS < R3, then GO TO 51

Make sure the slope between points #1 and #2 is not a vertical line, and if it is, then check for a solution.

29. If /X2-X1/ < 10^{-8}, and XS > X2, then GO TO 51

Make sure the slope between points #1 and #2 is not a vertical line.

30. If /X2-X1/ < 10^{-8}, then GO TO 35

Calculate the slope of the characteristic between points #1 and #2

31. SL12 = (R2-R1)/(X2-X1)

Determine the y-intercept of this characteristic

32. B12 = R1-(SL12)(X1)

Determine the y-intercept of the point using the same slope.

33. BP = RS-(SL12)(XS)

See if a solution is possible; if not try another set of points.

34. If BP > B12, then GO TO 51

A solution is still possible. Now determine whether the characteristic between points #2 and #3 is a vertical line. If so, then check for a solution.

35. If /X2-X3/ < 10^{-8} and XS > X2, then GO TO 51

Check for a vertical slope of this characteristic

36. If /X2-X3/ < 10^{-8}, then GO TO 42

Calculate the slope of the characteristic between points #2 and #3 and determine its y-intercept.

37. SL23 = (R2-R3)/(X2-X3)

38. B23 = R2 - (SL23)(X2)

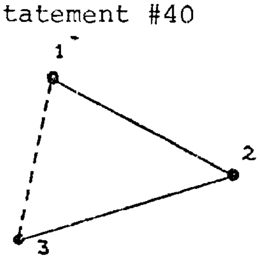
Determine the y-intercept of the point using the same slope

39. BP = RS - (SL23)(XS)

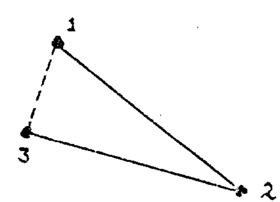
Check to see if a solution is still possible; if not, try another set of points.

40. If BP < B23 and SL23 > 0, then GO TO 51

Statement #40



Statement #41



41. If BP > B23 and SL23 < 0, then GO TO 51

Check for a vertical line between points #1 and #3, and if it is, then check for a solution.

42. If /X3-X1/ < 10^{-8} and XS < X3, then GO TO 51

Check for a vertical line between points #1 and #3. If this is the case, a solution has been reached.

43. If /X3-X1/ < 10^{-8}, then GO TO 49

Calculate the slope and y-intercept of the line between points #1 and #3.

44. SL31 = (R3-R1)/(X3-X1)

45. B31 = R1 - (SL31)(X1)

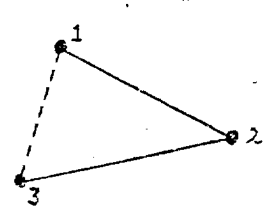
Determine the y-intercept of the points using the same slope.

46. BP = RS - (SL31)(XS)

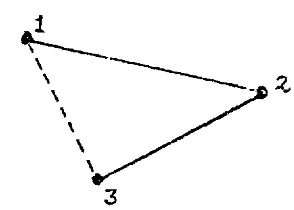
Test for a solution. If none is found, then try another set of points.

47. If BP > B31 and SL31 > 0, then GO TO 51

Statement #47



Statement #48



48. If BP < B31 and SL31 < 0, then GO TO 51

A solution has been reached. Reset the value of A1 indicating point #4 does not exist. Then return to STRLNE.

49. A4 = 0.0

50. RETURN

51. CONTINUE

Now the program will check for a solution at points in which four characteristic points will surround the given point.

First, set the starting value of the I-subscript for a completely filled characteristic matrix.

52. ISTT = 3

If the characteristics matrix is in the throat region and it was read in along a characteristic line then the starting value of the I-subscript is reset.

53. If LOC = 0, then ISTT = NDJ + 2

Begin incrementing the I-subscript

54. DO 129 I = ISTT, NDF

Determine whether I is odd or even

55. MI = MOD (I,2)

Determine the starting value of the J-subscript

56. JSTRT = 1

If I is an even number, reset the starting value of the J-subscript

57. If MI = 0, then JSTRT = 2

Begin incrementing the J-subscripts

58. DO 128 J = JSTRT, NDJ

Check for a partially filled characteristic matrix at various locations.

59. If R(I,J) < 0.0001 and LOC = 0, then GO TO 128

60. If R(I,J) < 0.0001 and ILOC = 3, then GO TO 128

61. If R(I,J) < 0.0001, then GO TO 128.

Go to another part of the program to calculate a lower boundary

62. If J = 1, then GO TO 131

The program will jump to the following card, so a check for a lower boundary point is made again.

63. If J = 1, then GO TO 128

Go to another part of the program to calculate an upper boundary

64. If J = NDJ and MI > 0, then GO TO 167

The program will jump to the following statement, so a check for an upper boundary is made again.

65. If J = NDJ and MI > 0, then GO TO 129

Set the four characteristic points. First see whether I is even. If so, points #1 and #3 will be set differently.

66. If MI = 0, then GO TO 77

The subscript I is odd.

67. X1 = X(I-1,J+1)

68. R1 = R(I-1,J+1)

Check again for a partially filled characteristic matrix

69. If X1 < 0.0001 and R1 < 0.0001, then GO TO 128

70. T1 = T(I-1,J+1)

71. NU1 = NU(I-1,J+1)

72. X3 = X(I-1,J)

73. R3 = R(I-1,J)

74. NU3 = NU(I-1,J)

75. T3 = T(I-1,J)

76. GO TO 85

The subscript I is even.

77. X1 = X(I-1,J)

78. R1 = R(I-1,J)

79. NU1 = NU(I-1,J)

80. T1 = T(I-1,J)

81. X3 = X(I-1,J-1)

82. R3 = R(I-1,J-1)

83. NU3 = NU(I-1,J-1)

84. T3 = T(I-1,J-1)

Points #2 and #4 will now be set.

85. X2 = X(I,J)

86. R2 = R(I,J)

87. NU2 = NU(I,J)

88. T2 = T(I,J)

89. X4 = X(I-2,J)

90. R4 = R(I-2,J)

91. NU4 = NU(I-2,J)

92. T4 = T(I-2,J)

Before checking the y-intercepts, see if the point has any possibility of being in this set of four points. If not, try another set of points.

93. If XS > X1 and XS > X2 and XS > X3 and XS > X4, then GO TO 128

94. If XS < X1 and XS < X2 and XS < X3 and XS < X4, then GO TO 128

95. If RS > R1 and RS > R2 and RS > R3 and RS > R4, then GO TO 128

96. If RS < R1 and RS < R2 and RS < R3 and RS < R4, then GO TO 128

Check for a vertical characteristic between points #1 and #2. If it is there check for a possible solution.

97. If /X2-X1/ < 10^{-8} and XS > X2, then GO TO 128

Check for a vertical characteristic between points #1 and #2

98. If /X2-X1/ < 10^{-8}, then GO TO 104

Calculate the slope and y-intercept of the characteristic between points #1 and #2

99. SL12 = (R2-R1)/(X2-X1)

100. B12 = R2 - (SL12)(X2)

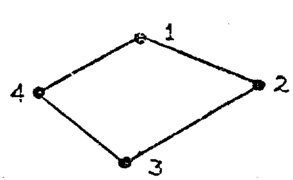
Calculate the y-intercept of the point using the same slope.

101. BP = RS - (SL12)(XS)

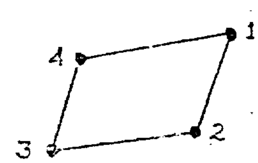
See if a solution is possible, if not try another set of points.

102. If SL12 < 0 and BP > B12, then GO TO 128

Statement #102



Statement #103



103. If SL12 > 0 and BP < B12, then GO TO 128

See if the characteristic between points #2 and #3 is vertical. If so, check for a solution.

104. If /X3-X2/ < 10^{-8} and XS > X2, then GO TO 128

Check for a vertical characteristic between points #2 and #3

105. If /X3-X2/ < 10^{-8}, then GO TO 112

Calculate the slope and y-intercept between points #2 and #3

106. SL23 = (R2-R3)/(X2-X3)

107. B23 = R2 - (SL23)(X2)

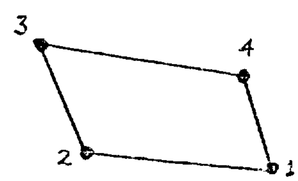
Determine the y-intercept of the point using the same slope.

108. BP = RS - (SL23)(XS)

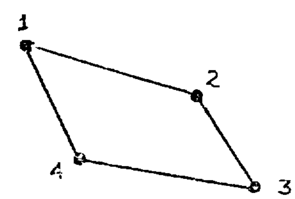
Check for a possible solution

109. If SL23 < 0 and X2 > X3 and BP < B23, then GO TO 128

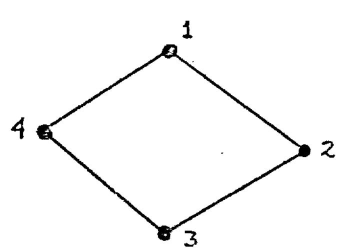
Statement #109



Statement #110



Statement #111



110. If SL23 < 0 and X3 > X2 and BP > B23, then GO TO 128

111. If BP < B23 and SL23 > 0, then GO TO 128

112. If $/X4-X3/ < 10^{-8}$ and $XS < X4$, then GO TO 128

Check for a vertical characteristic between points #3 and #4

113. If $/X4-X3/ < 10^{-8}$, then GO TO 119

Calculate the slope and y-intercept of the characteristic between points #3 and #4.

114. SL34 = (R3-R4)/(X3-X4)
B34 = R3 - (SL34)(X3)

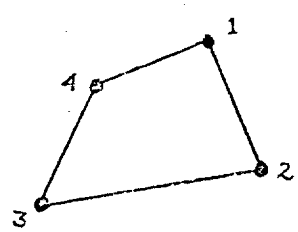
Calculate the y-intercept of the point using the same slope.

115. BP = RS - (SL34)(XS)

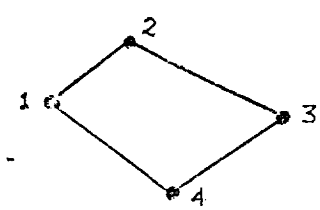
Check for a possible solution.

116. If SL34 > 0 and BP > B34 and X4 > X3, then GO TO 128

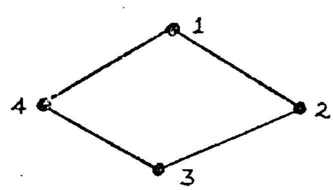
Statement #116



Statement #117



Statement #118



117. If SL34 > 0 and BP < B34 and X3 > X4, then GO TO 128

118. If SL34 < 0 and BP < B34, then GO TO 128

Check for a vertical characteristic between points #4 and #1. If this is the case then check for a solution.

119. If $/X4-X1/ < 10^{-8}$ and $XS < X4$, then GO TO 128

Check for a vertical characteristic between points #4 and #1. If this has occurred, a solution has been reached.

120. If $/X4-X1/ < 10^{-8}$, then GO TO 127

Calculate the slope and y-intercept of the characteristic between points #1 and #4.

121. SL41 = (R4-R1)/(X4-X1)

122. B41 = R1 - (SL41)(X1)

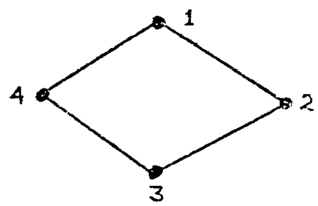
Calculate the y-intercept of the point using the same slope.

123. BP = RS - (SL41)(XS)

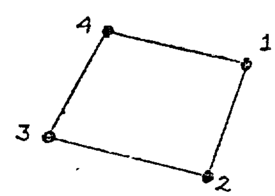
Check for a solution

124. If $BP > B41$ and $SL41 > 0$, then GO TO 128

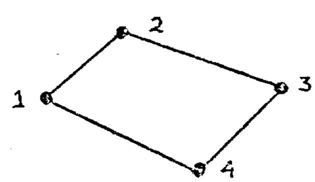
Statement #124



Statement #125



Statement #126



125. If $SL41 < 0$ and $X1 > X4$ and $BP > B41$, then GO TO 128

126. If $SL41 < 0$ and $X4 > X1$ and $BP < B41$, then GO TO 128

A solution has been reached

127. RETURN

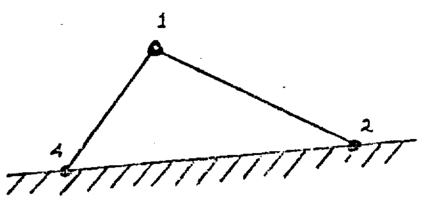
128. CONTINUE

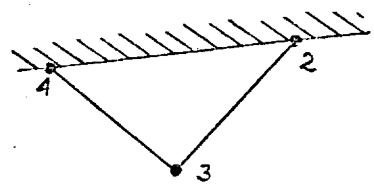
129. CONTINUE

Go to another part of the program to calculate the right hand side of the characteristics matrix.

130. GO TO 197

The part of the program considers the upper and lower edges of the characteristics matrix. For a lower surface, point #3 does not exist; for an upper surface, point #1 does not exist.





Set points #2 and #4.

131. X2 = X(I,J)

132. R2 = R(I,J)

133. NU2 = NU(I,J)

134. T2 = T(I,J)

135. X4 = X(I-2,J)

136. R4 = R(I-2,J)

137. NU4 = NU(I-2,J)

138. T4 = T(I-2,J)

See if it is an upper or lower surface.

139. If IPLC = 1, then GO TO 169

It is a lower surface; set point #1.

140. X1 = X(I-1,J+1)

141. R1 = R(I-1,J+1)

142. NU1 = NU(I-1,J+1)

143. T1 = T(I-1,J+1)

Before checking the y-intercepts, see if the point has any possibility of being in this set of three points. If not, move into the characteristic matrix.

144. If XS > X1 and XS > X2 and XS > X4, then GO TO 63

145. If XS < X1 and XS < X2 and XS < X4, then GO TO 63

146. If RS > R1 and RS > R2 and RS > R4, then GO TO 63

147. If RS < R1 and RS < R2 and RS < R4, then GO TO 63

Check to see if the characteristic between points #1 and #2 is vertical. If it is, then test for a possible solution.

148. If /X2-X1/ < 10^{-8}, and XS > X2, then GO TO 63

See if the characteristic between points #1 and #2 is vertical.

149. If /X2-X1/ < 10^{-8}, then GO TO 154

Calculate the slope and y-intercept of this characteristic line.

150. SL12 = (R1-R2)/(X1-X2)

151. B12 = R1-(SL12)(X1)

Calculate the y-intercept of the point using the same slope.

152. BP = RS - (SL12)(XS)

Test for a possible solution.

153. If BP > B12, then GO TO 63

Check to see if the characteristic between points #4 and #1 is vertical. If so, then test for a possible solution.

154. If /X1-X4/ < 10^{-8} and XS < X1, then GO TO 63

See if the characteristic between points #1 and #4 is vertical.

155. If /X1-X4/ < 10^{-8}, then GO TO 161

Calculate the slope and y-intercept of this characteristic line.

156. SL41 = (R4-R1)/(X4-X1)

157. B41 = R1 - (SL41)(X1)

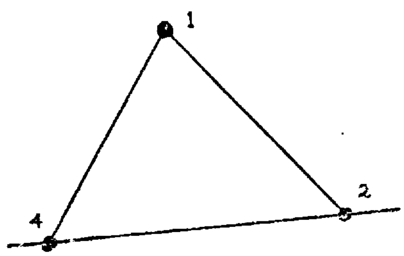
Calculate the y-intercept of the point using the same slope.

158. BP = RS - (SL41)(XS)

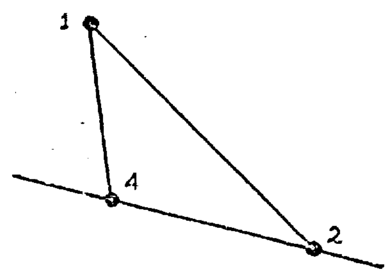
Test for a possible solution.

159. If BP > B41 and SL41 > 0, then GO TO 63

Statement #159



Statement #160



160. If BP < B41 and SL41 < 0, then GO TO 63

Calculate the slope and y-intercept of the line between points #2 and #4.

161. SL24 = (R2-R4)/(X2-X4)

162. B24 = R2 - (SL24)(X2)

Calculate the y-intercept of the point using the same slope.

163. BP = RS - (SL24)(XS)

Test for a solution.

164. If BP < B24, then GO TO 63

A solution has been reached. Set A3 = 0.0 denoting a solution on the lower characteristics matrix, and then return to STRLNE.

165. A3 = 0.00

166. RETURN

Set the integer denoting an upper surface.

167. IPLC = 1

Go to another part of the program to set points #2 and #4.

168. GO TO 131

Reset the integer

169. IPLC = 0

Set point #3

170. X3 = X(I-1,J)

171. R3 = R(I-1,J)

172. NU3 = NU(I-1,J)

173. T3 = T(I-1,J)

Before checking y-intercepts, see if the point has any possibility of being in this set of three points. If not, move back into the characteristic matrix.

174. If XS > X2 and XS > X3 and XS > X4, then GO TO 65

175. If XS < X2 and XS < X3 and XS < X4, then GO TO 65

176. If RS > R2 and RS > R3 and RS > R4, then GO TO 65

177. If RS < R2 and RS < R3 and RS < R4, then GO TO 65

Check if the characteristic between points #2 and #3 is a vertical line. If it is, then test for a possible solution.

178. If /X3-X2/ < 10^{-8} and XS > X3, then GO TO 65

See if the characteristic between points #2 and #3 is vertical.

179. If /X3-X2/ < 10^{-8}, then GO TO 185

Calculate the slope and y-intercept of the characteristic between points #2 and #3

180. SL23 = (R2-R3)/(X2-X3)

181. B23 = R2 - (SL23)(X2)

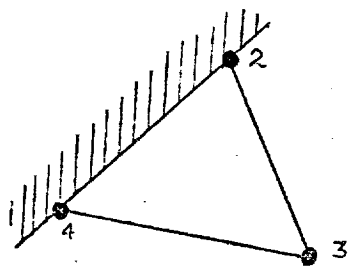
Determine the y-intercept of the point using the same slope.

182. BP = RS - (SL23)(XS)

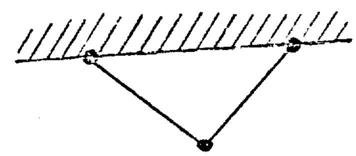
Test for a possible solution.

183. If BP > B23 and SL23 < 0, then GO TO 65

Statement #183



Statement #184



184. If BP < B23 and SL23 > 0, then GO TO 65

Check if the characteristic between points #3 and #4 is a vertical line. If so, then check for a possible solution.

185. If /X3-X4/ < 10^{-8} and XS < X3, then GO TO 65

See if the characteristic between points #2 and #3 is vertical

186. If /X3-X4/ < 10^{-8}, then GO TO 191

Calculate the slope and y-intercept of the characteristic between points #2 and #3.

187. SL34 = (R3-R4)/(X3-X4)

188. B34 = R3 - (SL34)(X3)

Calculate the y-intercept of the point using the same slope.

189. BP = RS - (SL34)(XS)

Test for a possible solution.

190. If BP < B34, then GO TO 65

Calculate the slope and y-intercept of the line between points #4 and #2.

191. SL42 = (R4-R2)/(X4-X2)

192. B42 = R4 - (SL42)(X4)

Calculate the y-intercept of the point using the same slope.

193. BP = RS - (SL42)(XS)

Test for a solution

194. If BP > B42, then GO TO 65

A solution has been reached. Set A1 = 0.0 indicating a solution at the top of the characteristic matrix, and then return to STRLNE.

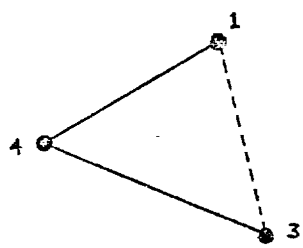
195. A1 = 0.00

196. RETURN

See if recompression has occurred. If so, the solution has not been found.

197. If ILOC = 3, then GO TO 241

This last section of the program will take care of the possibility of a solution along the right edge of the characteristics matrix. For this situation, point #2 does not exist.



Set the J-subscript to move up this row.

198. DO240 J = 2,NDJ

Set the points.

199. X1 = X(NDF,J)

200. R1 = R(NDF,J)

201. NU1 = NU(NDF,J)

202. T1 = T(NDF,J)

203. X2 = 0.0

204. R2 = 0.0

205. NU2 = 0.0

206. T2 = 0.0

207. X3 = X(NDF,J-1)

208. R3 = R(NDF,J-1)

209. NU3 = NU(NDF,J-1)

210. T3 = T(NDF,J-1)

211. X4 = X(NDF-1,J)

212. R4 = R(NDF-1,J)

213. NU4 = NU(NDF-1,J)

214. T4 = T(NDF-1,J)

Before calculating the y-intercepts, determine if the point has any possibility of being in this set of three points. If not try another set of three points.

215. If XS > X1 and XS > X3 and XS > X4, then GO TO 240

216. If XS < X1 and XS < X3 and XS < X4, then GO TO 240

217. If RS > R1 and RS > R3 and RS > R4, then GO TO 240

218. If RS < R1 and RS < R3 and RS < R4, then GO TO 240

See if the line between points #1 and #3 is vertical. If so, then test for a possible solution.

219. If /X3-X1/ < 10^{-8} and XS > X3, then GO TO 240

See if the line between points #1 and #3 is vertical.

220. If /X3-X1/ < 10^{-8}, then GO TO 226

Calculate the slope and y-intercept of the line between points #1 and #3.

221. SL13 = (R3-R1)/(X3-X1)

222. B13 = R1 - (SL13)(X1)

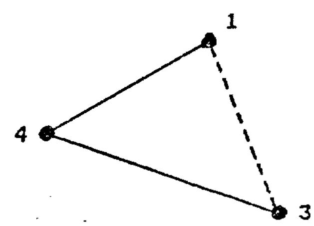
Calculate the y-intercept of the point using the same slope.

223. BP = RS - (SL13)(XS)

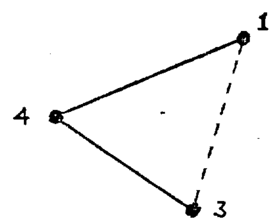
Test for a possible solution.

224. If BP > B13 and SL13 < 0, then GO TO 240

Statement # 224



Statement # 225



225. If BP < B13 and SL13 > 0, then GO TO 240

Check if the characteristic between points #3 and #4 is vertical. If so, see if a possible solution exists.

226. If /X3-X4/ < 10^{-8} and XS < X3, then GO TO 240

See if the characteristic between points #3 and #4 is vertical.

227. If /X3-X4/ < 10^{-8}, then GO TO 232

Calculate the slope and y-intercept of the characteristic between points #3 and #4.

228. SL34 = (R3-R4)/(X3-X4)

229. B34 = R3 - (SL34)(X3)

Calculate the y-intercept of the point using the same slope.

230. BP = RS - (SL34)(XS)

Check for a possible solution

231. If BP < B34, then GO TO 240

See if the characteristic between points #4 and #1 is vertical. If it is, then test for a solution.

232. If /X4-X1/ < 10^{-8} and XS < X4, then GO TO 240

See if the characteristic between points #4 and #1 is vertical.

233. If /X4-X1/ < 10^{-8}, then GO TO 238

Calculate the slope and y-intercept of the characteristic between points #4 and #1.

234. SL41 = (R4-R1)/(X4-X1)

235. B41 = R4 - (SL41)(X4)

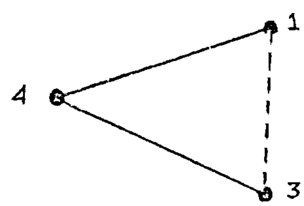
Calculate the y-intercept of the point using the same slope.

236. BP = RS - (SL41)(XS)

Test for a solution.

237. If BP > B41, then GO TO 240

Statement #237



A solution has been obtained. Set A2 = 0 indicating a solution on the right side of the characteristics matrix, and then return to STRLNE.

238. A2 = 0.00

239. RETURN

240. CONTINUE

A solution has not been found in the characteristics matrix. Set the integer noting this, and return to the statement number designated by STRLNE.

241. IEND = 1

242. RETURN 24

243. END

XLII. SUBROUTINE LIPSHK

XLII. SUBROUTINE LIPSHK

Subroutine LIPSHK calculates the effects of a "lip shock" originating at the tip of the plug. This shock arises from the possibility that the flow from the surface of the plug may undergo a compression rather than an expansion. (See Fig. XLII-1 below)

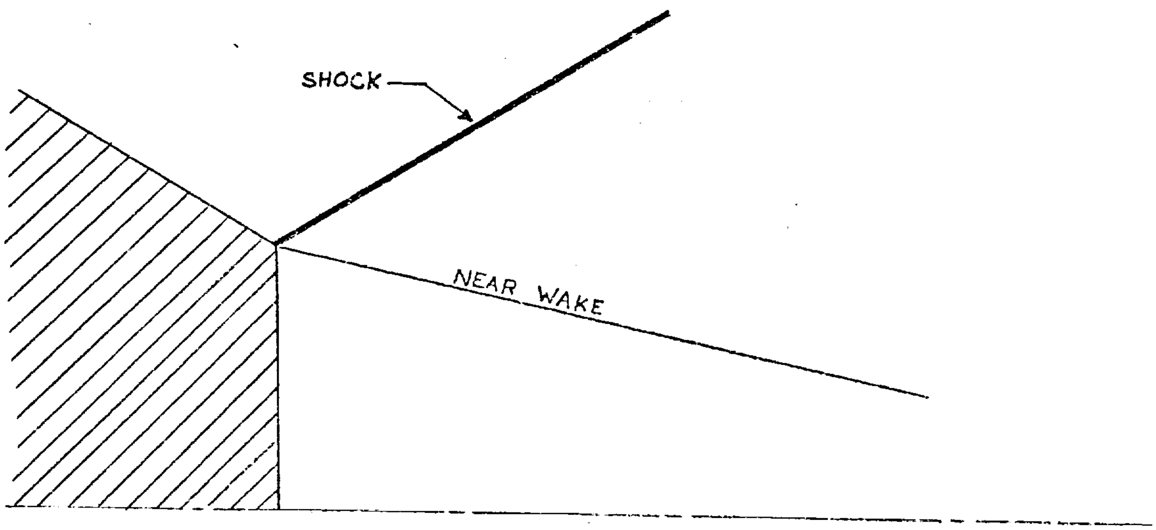


FIGURE XLII-1

This subroutine is similar in parts to the subroutines SSHAPE, FBASE6, and FLOW in that many of the calculations need to be repeated here because of the location of the shock. The possibility that both the "lip shock" and recompression shock are in the same characteristic matrix is also considered.

COMMON BLOCKS

COMMON blocks AMB, BLK3A, CNRANG, CORNER, DATBLK, GAS, PARAM, PLCBLK, POLIP, PTNOS, SIZE, SKLIP, SOLBLK, STRBLK, STRL, STRSRT, THETBK and TPN are used.

TPNZZL SUBROUTINES

Subroutines and function subprograms TPNZZL, BASE6, and FBASE6 call LIPSHK

LIPSHK uses subroutines and function subprograms CALC, CPB, FLOW, OSHOCK, PMANGL, PMTURN, SHOCK, SSHAPE, STRLNE, SURF, SURFSK, and TAB

FORTRAN SYSTEM ROUTINES

Built-in FORTRAN functions ASIN, ATAN, MOD, SIN, SQRT and TAN are used.

CALLING SEQUENCE

CALL LIPSHK (IREADS, NEPS, $)

IREADS is the type of initial line read-in used

$$\text{NEPS} = \begin{cases} 1 \text{ for axisymmetric flow} \\ 0 \text{ for planar flow} \end{cases}$$

ACST4 - accuracy requirement used in locating the j-streamline

AMBM - Mach number downstream of the "lip shock" - free jet boundary intersection

ANGCH - angle of a characteristic wave

COTDEL - the cotangent of the change in streamline angle across a shock

CPNUY - Prandtl-Meyer angle (radians) corresponding to AMBM

C3AS - square of the Crocco number just upstream of recompression

D - non-dimensional distance along the "lip shock"

DD - incremental non-dimensional distance along the shock

DDEL - minimum change in streamline angle (radians) across the shock

DEL - change in streamline angle (radians) across a shock

DELP - equal to 90% of the distance along the shock

DELTA - change in streamline angle (radians) across a shock

DFMY - difference in Mach number downstream of the shock as calculated from the shock equation and from the method of characteristics.

DJM1 - real number (= NDJ-1)

DNU - difference in Prandtl-Meyer angle from the ambient value to the downstream shock value (radians)

DRS - density ratio across the shock

DS - non-dimensional distance along the shock

DSR - ratio of the entropy change across the shock to the gas constant

DTC - incremental change in streamline angle in going from the plug surface to the near wake.

DTOT - total non-dimensional distance along the shock

D4 - one quarter of the total number of J-subscripted characteristics

D4M1 - D4 - 1

D75 - three quarters of the total number of J-subscripted characteristics

EKJ - variable used in determining the J-subscript

EMU2 - Mach angle (radians) immediately after separation

EMX - Mach number on the upstream side of the shock

EM1 - Mach number just before separation

EM2 - Mach number immediately after separation

G - ratio of specific heats

I - subscript

IBOUND - integer denoting the type of near wake solution desired

IDIFF - number of I-subscripts between recompression and the end of the characteristics matrix

ID1 - IDIFF + 1

ID2 - IDIFF + 2

ID3 - IDIFF + 3

IEND - final value of the I-subscript

IFIN - final value of the I-subscript

IINT - the I-subscript at an intersection

IK - subscript

IL - subscript

ILAST - final value of the I-subscript

ILOCN - integer denoting which part of the flow field is being calculated

IM - subscript

IP1 - subscript

IREC - value of the I-subscript at recompression

IRECP - integer denoting whether recompression has been encountered

IS - starting value of the I-subscript

IST - starting value of the I-subscript

IS1 - subscript

J - subscript

JFIN - final value of the J-subscript

JK - subscript

JLAST - final value of the J-subscript

JP - subscript

JS - subscript

JST - starting value of the J-subscript

JS1 - subscript

JTOP - final value of the J-subscript

JX - subscript

J1 - subscript

K - subscript

KK - subscript

K1 - subscript

L - subscript

LOCN - integer denoting which part of the flow field is being calculated

LS - subscript

LSHK - integer denoting that a "lip shock" is present

LTP - subscript

MAMB - Mach number on the free jet boundary upstream of the "lip shock" intersection with this boundary

MESHPM - factor used to set the minimum number of discrete turns when expanding from the plug surface to the near wake

MI - integer used to determine whether I is odd or even

MINT - upstream Mach number at a shock point

MS - Mach number at an upstream point on the shock

MSTR - array of Mach numbers along each streamline

MY - Mach number at a downstream shock point

M1A - Mach number just before separation

M2A - Mach number just after separation

M3A - Mach number just before recompression

NCOUNT - integer used to count iterations

NDF - final value of the I-subscript in the charactersitics matrix

NDI - maximum number of I-subscripts in the characteristics matrix

NDIM1 - NDI-1

NDIM2 - NDI-2

NDIP - equivalent to NDI

NDJ - maximum number of J-subscripts in the characteristics matrix

NDJM1 - NDJ-1

NDJP - equivalent to NDJ

NDJ2 - NDJ/2

ND4 - NDJ/4

ND75 - integer equivalent to D75

NINTS - number of characteristic-shock intersections

NOIPTS - number of points on the initial line

NOPPTS - number of plug surface coordinate points

NOPS - number of J-subscripts along a shock

NOSPTS - maximum number of streamline points

NOSTRL - number of streamlines

NP1 - NOPS-1

NS - Prandtl-Meyer angle (radians) on the upstream side of the shock (an array)

NSKPTS - number of shock points

NSL1 - NOSTRL - 1

NSPTS - equivalent to NOSPTS

NSTRT - starting value of the subscript on each streamline

NSTT - equivalent to NSTRT

NS1 - array equivalent to NUSX1

NTC - number of discrete turns in expanding from the plug surface to the near wake

NU - array of Prandtl-Meyer angles (radians) at each point in the characteristic matrix

NUAMB - Prandtl-Meyer angle (radians) corresponding to MAMB

NUINT - Prandtl-Meyer angle at an intersection (radians)

NUSX - array of Prandtl-Meyer angles (radians) at each point along the upstream side of the shock

NUSX1 - array of Prandtl-Meyer angles (radians) at the first point upstream of the shock

NUSY - array of Prandtl-Meyer angles (radians) at each point along the downstream side of the shock

NUX - array equivalent to NUSX

NUY - array equivalent to NUSY

N1 - NSKPTS +1

PA - ambient pressure (lb/in^2)

POLP - stagnation pressure (lb/in^2) downstream of the "lip shock" at its intersection with the free jet boundary

POR - ratio of stagnation pressures across a shock

POY - equivalent to POLP

POYPOX - equivalent to POR

PO1 - chamber stagnation pressure (lb/in^2)

PR - static pressure ratio across the shock

PYPX - equivalent to PR

P1 - static pressure (lb/in^2) just ahead of separation

R - array of non-dimensional radial coordinates at each point in the characteristics matrix.

RATIO - a ratio of distances used for interpolation

RB - non-dimensional radial coordinate of the streamline point just upstream of the shock-streamline intersection

RDIM - radial coordinate (inches) of a shock point (used for output)

RG - gas constant ($ft\text{-}lb_f/lb_m\text{-}^oR$)

RINT - non-dimensional radial coordinate at an intersection

RP - array of non-dimensional radial coordinates at each point along the surface of the plug

RPB - plug base radius (inches)

RQ - the non-dimensional radial coordinate of the point whose characteristic intersects the shock

RREC - the non-dimensional radial coordinate at recompression

RS - equivalent to RSX

RSRB - wake radius ratio

RSTR - array of non-dimensional radial coordinates at each point along a streamline

RSX - array of non-dimensional radial coordinates at each point along the upstream side of the shock

RSX1 - array of non-dimensional radial coordinates at the first point upstream of the shock

RS1 - array equivalent to RSX1

RYRX - density ratio across the shock

R2 - non-dimension plug base radius, OR the non-dimensional radius at recompression

S - array of entropies ($ft\text{-}lb_f/lb_m\text{-}^oR$) at each point in the characteristics matrix

SA - array of shock wave angles (radians) at each point along the shock

SACALC - shock wave angle (radians) at the intersection of a streamline and the shock wave

SAR - shock wave angle (radians) at a point

SAS - array equivalent to SA

SINT - the entropy (ft-lb_f/lb_m-^oR) at an intersection

SLB - slope of the free jet boundary

SLCH - slope of a characteristic wave

SLSK - slope of a shock at a point

SLST - slope of a streamline at a point

SM - square of the Mach number at an intersection

SOD - sum of the non-dimensional distances along the shock

SOL3A - the ratio of the Mach number at recompression to that after separation, OR the streamline angle (radians) at recompression

SP - the entropy (ft-lb_f/lb_m-^oR) on the plug surface

SS - array equivalent to SXS

SSX - array of entropies (ft-lb_f/lb_m-^oR) at each point along the upstream side of the shock

SSX1 - array of entropies (ft-lb_f/lb_m-^oR) at the first point upstream of the shock

SSY - array of entropies (ft-lb_f/lb_m-^oR) at each point along the downstream side of the shock

SS1 - array equivalent to SSX1

SW - entropy (ft-lb_f/lb_m-^oR) along the near wake surface

SWAD - shock wave angle (degrees) (used for output)

SWAR - shock wave angle (radians) at a point

SX - upstream entropy (ft-lb_f/lb_m-^oR) at the intersection of a characteristic and a shock

SY - downstream entropy (ft-lb_f/lb_m-^oR) at the intersection of a characteristic and the shock

S3A - the entropy (ft-lb_f/lb_m-^oR) just ahead of recompression

S3NT - entropy (ft-lb_f/lb_m-^oR) at the point where the characteristic intersects the shock

T - array of streamline angles (radians) at each point in the characteristic matrix

THET12 - change in streamline angle (radians) in going from the plug surface to the near wake

THET3A - streamline angle (radians) at recompression

TINT - streamline angle (radians) at an intersection

TO1 - chamber stagnation temperature (^oR)

TP - array of streamline angles (radians) at each point on the plug surface

TRS - static temperature ratio across the shock

TS - array equivalent to TSX

TSTR - array of streamline angles (radians) at each point along a streamline

TSX - array of streamline angles (radians) at each point along the upstream side of the shock

TSX1 - array of streamline angles (radians) at the first point upstream of the shock

TSY - array of streamline angles (radians) at each point along the downstream side of the shock

TS1 - array equivalent to TSX1

TW - streamline angle (radians) of a conical near wake

TX - streamline angle (radians) at the intersection of a shock and a characteristic

TXD - streamline angle (degrees) at an upstream point on the shock (used for output)

TY - downstream streamline angle (radians) at the intersection of a shock and characteristic

TYD - streamline angle (degrees) at a downstream point on the shock (used for output)

TYG - estimate of the downstream streamline angle (radians) at the intersection of a shock and a characteristic

TYTX - static temperature ratio across a shock

T2 - streamline angle (radians) of the last plug surface point

T3A - streamline angle (radians) just upstream of recompression

V3A - Prandtl-Meyer angle (radians) just upstream of recompression

X - array of non-dimensional axial coordinates at each point in the characteristics matrix

XB - non-dimensional axial coordinate of the streamline point just upstream of the shock-streamline intersection

XCALC - non-dimensional axial coordinate near the intersection of the shock wave and a streamline

XDIM - axial coordinate (inches) of a shock point (used for output)

XINT - non-dimensional axial coordinate at an intersection

XP - array of non-dimensional axial coordinates at each point on the plug surface

XPF - non-dimensional axial coordinate of the plug base

XQ - non-dimensional axial coordinate of the point whose characteristic intersects the shock

XREC - non-dimensional axial location of recompression

XS - equivalent to XSX (an array)

XSTR - array of non-dimensional axial coordinates at each point along a streamline

XSX - array of non-dimensional axial coordinates at each point along the upstream side of the shock

XSX1 - array of non-dimensional axial coordinates at the first point upstream of the shock

XS1 - array equivalent to XSX1

X2 - non-dimensional axial location of the plug base, OR the non-dimensional axial location of recompression

SOLUTION METHOD

Redefine the entropy at the boundary

1. SP = S(1,1)

Define calculational integers

2. NDIM1 = NDI-1

3. NDIM2 = NDI-2

4. NDJM1 = NDJ-1

Change this last integer to a real number

5. DJM1 = NDJM1

See if a final solution has been obtained. If so, go to another part of the program and print out the final "lip shock" shape.

6. If IREADS = 6, then GO TO 614

See if the "lip shock" and the recompression shock are in the same characteristic matrix

7. If IREADS = 4, then GO TO 555

Set the first shock point at the tip of the plug base

8. XSX(1) = XP(NOPPTS)

9. RSX(1) = RP(NOPPTS)

10. NUSX(1) = PMANGL(M1A,G)

11. TSX(1) = T2

12. SSX(1) = SP

Determine the compressive turn from the plug surface to the near wake.

13. DELTA = -THET12

Set a minimum change in streamline angle (radians) across the shock.

14. DDEL = 0.0100

Calculate the Mach number at this first point upstream of the shock.

15. CALL PMTURN

Calculate the shock wave angle of the shock and determine the downstream variables at this first point.

16. CALL SHOCK

17. CALL OSHOCK

18. M2A = MY

19. SW = SP + (DSR)(RG)

20. TSY(1) = T2 + DELTA

21. NUSY(1) = PMANGL(MY,G)

22. SSY(1) = SW

Set the starting value of the number of shock points.

23. K = 2

This next portion of LIPSHK sets up the initial estimate of the "lip shock" shape.

24. DO 74 L = 2, NDJ

Determine the upstream Mach number and shock wave angle at a point on the shock.

25. CALL PMTURN

26. CALL SHOCK

Calculate the shock angle (radians) using the centerline as a reference coordinate, and calculate its slope.

27. SA(K-1) = SWAR + TSX(K-1)

28. SLSK = TAN(SA(K-1))

Set a real number used in determining the J-subscript

29. EKJ = L + 0.60

Search along each characteristic for an intersection of a characteristic with the projected shock wave.

30. DO 73 I = 2, L

Increment the real number and set the J-subscript.

31. EKJ = EKJ - 0.50

32. J = EKJ

See if the characteristics matrix is filled.

33. If $X(I,J) < 10^{-6}$, then GO TO 74

See whether the subscript I is odd or even.

34. MI = MOD(I,2)

35. If MI = 0, then GO TO 47

I is an odd number. Calculate the slope of the characteristic and its intersection with the shock.

36. SLCH = (R(I,J)-R(I-1,J+1))/(X(I,J)-X(I-1,J+1))

37. XINT = (R(I-1,J+1)-RSX(K-1)+(SLSK)(XSX(K-1))-
(SLCH)(X(I-1,J+1)))/(SLSK-SLCH)

Now test to see whether this intersection is actually between two characteristic points. If not, try two more points.

38. If X(I-1,J+1) > XINT, then GO TO 74

39. If XINT > X(I,J), then GO TO 73

40. If XINT < XSX(K-1), then GO TO 74

An intersection has been found. Save the variables at the characteristics point upstream of the shock.

41. XSX1(K) = X(I-1,J+1)

42. RSX1(K) = R(I-1,J+1)

43. NUSX1(K) = (NU(I-1,J+1)

44. TSX1(K) = T(I-1,J+1)

45. SSX1(K) = S(I-1,J+1)

Go to another part of the program to calculate the variables on the shock itself.

46. GO TO 57

I is an even number. Calculate the slope of the characteristic and its intersection with the shock.

47. SLCH = (R(I,J) - R(I-1,J))/(X(I,J) - X(I-1,J))

48. XINT = (R(I-1,J) - RSX(K-1) + (SLSK)(XSX(K-1)) -
(SLCH)(X(I-1,J)))/(SLSK-SLCH)

Now test to see whether the intersection is actually between these two characteristic points. If not try two more points.

49. If X(I-1,J) > XINT, then GO TO 74

50. If XINT > X(I,J), then GO TO 73

51. If XINT < XSX(K-1), then GO TO 74

An intersection has been found. Save the variables at the characteristic point upstream of the shock.

52. XSX1(K) = X(I-1,J)

53. RSX1(K) = R(I-1,J)

54. NUSX1(K) = NU(I-1,J)

55. TSX1(K) = T(I-1,J)

56. SSX1(K) = S(I-1,J)

Now set the upstream variables at the actual intersection of the characteristic and the shock.

57. XSX(K) = XINT

58. RSX(K) = (SLCH)(XINT-X(I,J)) + R(I,J)

Define a ratio of distances and calculate remaining variables.

59. RATIO = (XINT-XSX(K))/(X(I,J)-XSX1(K))

60. NUSX(K) = NUSX1(K)+(RATIO)(NU(I,J)-NUSX1(K))

61. TSX(K) = TSX1(K)+(RATIO)(T(I,J)-TSX1(K))

62. SSX(K) = SSX1(K)+(RATIO)(S(I,J)-SSX1(K))

Calculate the streamline angle on the downstream side of the shock.

63. TSY(K) = TSX(K) + DELTA

Determine the change in streamline angle through the shock, and make sure it is greater than the minimum angle.

64. DEL = TSY(K) - TSX(K)

65. If DEL < DDEL, then TSY(K) = TSX(K) + DDEL

Determine the Mach number at the upstream intersection

66. CALL PMTURN

Reset the change in streamline angle across the shock and then determine the shock angle and set the variables on the downstream side of the shock.

67. DELTA = TSY(K) - TSX(K)

68. CALL SHOCK

69. CALL OSHOCK

70. NUSY(K) = PMANGL(MY,G)

71. SSY(K) = (DSR)(RG) + SSX(K)

Increase the number of shock points by one.

72. K = K + 1

73. CONTINUE

74. CONTINUE

Continue looking for intersections between the shock and the characteristics whose right-running waves begin at the free jet boundary. First, set the last value of the I-subscript, and then begin searching.

75. IFIN = NDJ

76. DO 126 L = 4, NDIM2, 2

Calculate the upstream Mach number and shock wave angle of the previous shock point.

77. CALL PMTURN

78. CALL SHOCK

Determine the shock angle using the centerline as the reference coordinate, and then calculate its slope.

79. SA(K-1) = SWAR + TSX(K-1)

80. SLSK = TAN(SA(K-1))

Set a real number used to determine the J-subscript.

81. EKJ = NDJ + 0.60

Begin searching for an intersection

82. DO 125 I = L, IFIN

Set the J-subscript.

83. EKJ = EKJ - 0.50

84. J = EKJ

See if the characteristic matrix is filled.

85. If X(I,J) < 10^{-6}, then GO TO 126

See if the subscript I is odd or even.

86. MI = MOD(I,2)

87. If MI = 0, then GO TO 99

I is an odd number. Calculate the slope of the characteristic and its intersection with the shock.

88. SLCH = (R(I,J) - R(I-1,J+1))/(X(I,J) - X(I-1,J+1))

89. XINT = (R(I-1,J+1) - RSX(K-1) + (SLSK)(XSX(K-1)) - (SLCH)(X(I-1,J+1)))/(SLSK-SLCH)

Now test to see whether this intersection is actually between these two characteristic points. If not, try two more points.

90. If X(I-1,J+1) > XINT, then GO TO 126

91. If XINT > X(I,J), then GO TO 125

92. If XINT < XSX(K-1), then GO TO 126

An intersection has been found. Save the variables at the characteristic point upstream of the shock.

93. XSX1(K) = X(I-1,J+1)

94. RSX1(K) = R(I-1,J+1)

95. NUSX1(K) = NU(I-1,J+1)

96. TSX1(K) = T(I-1,J+,)

97. SSX1(K) = S(I-1,J+1)

Go to another part of the program to set the variables at the intersection of the shock and the characteristic.

98. GO TO 109

I is an even number. Calculate the slope of the characteristic and its intersection with the shock.

99. SLCH = (R(I,J) - R(I-1,J))/(X(I,J) - X(I-1,J))

100. XINT = (R(I-1,J) - RSX(K-1) + (SLSK)(XSX(K-1)) -
(SLCH)(X(I-1,J)))/(SLSK-SLCH)

Now test to see whether this intersection is actually between these two characteristic points. If not, try two more points.

101. If X(I-1,J) > XINT, then GO TO 126

102. If XINT > X(I,J), then GO TO 125

103. If XINT < XSX(K-1), then GO TO 126

An intersection has been found. Save the variables at the characteristic point upstream of the shock.

104. XSX1(K) = X(I-1,J)

105. RSX1(K) = R(I-1,J)

106. NUSX1(K) = NU(I-1,J)

107. TSX1(K) = T(I-1,J)

108. SSX1(K) = S(I-1,J)

Now set the upstream variables at the actual intersection of the characteristic and the shock.

109. XSX(K) = XINT

110. RSX(K) = (SLCH)(XINT - X(I,J)) + R(I,J)

Define a ratio of distances, and calculate the remaining upstream variables.

111. RATIO = (XINT-XSX1(K))/(X(I,J) - XSX1(K))

112. NUSX(K) = NUSX1(K) + (RATIO)(NU(I,J) - NUSX1(K))

113. TSX(K) = TSX1(K) + (RATIO)(T(I,J) - TSX1(K))

114. SSX(K) = SSX1(K) + (RATIO)(S(I,J) - SSX1(K))

Calculate the streamline angle on the downstream side of the shock

115. TSY(K) = TSX(K) + DELTA

Determine the change in streamline angle through the shock, and make sure it is greater than the minimum angle.

116. DEL = TSY(K) - TSX(K)

117. If DEL < DDEL, then TSY(K) = TSX(K) + DDEL

Determine the Mach number at the upstream intersection

118. CALL PMTURN

Reset the change in streamline angle across the shock, and then determine the shock angle, and set the remaining variables on the downstream side of the shock.

119. DELTA = TSY(K) - TSX(K)

120. CALL SHOCK

121. CALL OSHOCK

122. NUSY(K) = PMANGL(MY,G)

123. SSY(K) = (DSR)(RG) + SSX(K)

Increase the number of shock points by one.

124. K = K + 1

125. CONTINUE

126. CONTINUE

The initial estimate of the "lip shock" shape has now been set. Determine the total number of shock points.

127. NSKPTS = K - 1

This next section determines the intersection of the "lip shock" with the external free jet boundary. First set the J-subscript.

128. J = NDJ

Calculate the shock wave angle and its slope.

129. SA(K-1) = SWAR + TSX(K-1)

130. SLSK = TAN(SA(K-1))

Begin moving along the top surface of the characteristic matrix looking for an intersection.

131. DO 161 I = 3, NDIM1, 2

Look for an incomplete characteristic matrix

132. If X(I,J) < 10^{-6}, then GO TO 162

Calculate the slope of the boundary and determine its intersection with the shock.

133. SLCH = (R(I,J) - R(I-1,J))/(X(I,J) - X(I-1,J))

134. XINT = (R(I,J) - RSX(K-1) + (SLSK)(XSX(K-1)) -
(SLCH)(X(I,J)))/(SLSK-SLCH)

Test for an actual intersection between the two characteristic points. If not, try two more points.

135. If X(I-1,J) > XINT, then GO TO 162

136. If XINT > X(I,J), then GO TO 161

Save the point and its variables at the intersection and at the characteristic point upstream of the intersection.

137. XSX(K) = XINT

138. XSX1(K) = X(I-2,J)

139. RSX1(K) = R(I-2,J)

140. NUSX1(K) = NU(I-2,J)

141. TSX1(K) = T(I-2,J)

142. SSX(K) = S(I-2,J)

143. RSX(K) = (SLCH)(XINT-X(I,J)) + R(I,J)

Calculate a ratio of distances, and then determine the remainder of the upstream variables.

144. RATIO = (XINT-XSX1(K))/(X(I,J) - XSX1(K))

145. NUSX(K) = NUSX1(K) + (RATIO)(NU(I,J)- NUSX1(K))

146. TSX(K) = TSX1(K) + (RATIO)(T(I,J) - TSX1(K))

147. SSX(K) = SSX1(K) + (RATIO)(S(I,J) - SSX1(K))

Determine the upstream Mach number at the intersection, and then calculate the downstream properties.

148. CALL PMTURN

149. CALL OSHOCK

Calculate the stagnation pressure downstream of the "lip shock" at this point, and also the static pressure ratio.

150. POY = (POR)(PO1)

151. PR = POY/PA

Calculate the Mach number corresponding to this static pressure ratio and its equivalent Prandtl-Meyer angle.

152. $AMBM = (2)PR^{\frac{G-1}{G}} - 1)/(G-1))^{\frac{1}{2}}$

153. NUAMB = PMANGL(AMBM,G)

Calculate the Prandtl-Meyer angle of the downstream Mach number resulting from the shock wave solution.

154. CPNUY = PMANGL (MY,G)

Determine the difference between these two values of the Prandtl-Meyer angle and set the downstream variables at this point.

155. DNU = NUAMB - CPNUY

156. TSY(K) = DNU

157. NUSY(K) = NUAMB

158. SSY(K) = (DSR)(RG) + SSX(K)

Reset the total number of shock points and stop looking for any further points.

159. NSKPTS = K

160. GO TO 162

161. CONTINUE

Make sure there are enough intersections to calculate a "lip shock". If not, then return to the calling program.

162. If NSKPTS < 2, then LSHK = 0

163. If LSHK = 0, then RETURN

Redefine the first upstream point on the plug boundary as the intersection point itself.

164. XSX1(1) = XSX(1)

165. RSX1(1) = RSX(1)

166. NUSX1(1) = NUSX(1)

167. TSX1(1) = TSX(1)

168. SSX1(1) = SSX(1)

Redefine the arrays of variables along the shock.

169. DO 180 LS = 1,NSKPTS

170. XS(LS) = XSX(LS)

171. RS(LS) = RSX(LS)

172. TS(LS) = TSX(LS)

173. NS(LS) = NUSX(LS)

174. SS(LS) = SSX(LS)

175. SAS(LS) = SA(LS)

176. XS1(LS) = XSX1(LS)

177. RS1(LS) = RSX1(LS)

178. TS1(LS) = TSX1(LS)

179. NS1(LS) = NUSX1(LS)

180. SS1(LS) = SSX1(LS)

Now reset all (except at the tip of the plug) variables equal to zero.

181. DO195 LS = 2,NDI

182. XSX(LS) = 0.0

183. RSX(LS) = 0.0

184. TSX(LS) = 0.0

185. TSY(LS) = 0.0

186. NUSX(LS) = 0.0

187. NUSY(LS) = 0.0

188. SSX(LS) = 0.0

189. SSY(LS) = 0.0

190. XSX1(SL) = 0.0

191. RSX1(LS) = 0.0

192. NUSX1(LS) = 0.0

193. TSX1(LS) = 0.0

194. SSX1(LS) = 0.0

195. SA(LS) = 0.0

Determine the linear distance along the "lip shock". The distance will be measured from the point at the tip of the plug.

196. DS(1) = 0.0

197. DO 198 LS = 2, NSKPTS

198. $DS(LS) = DS(LS-1) + ((XS(LS) - XS(LS-1))^2 + (RS(LS) - RS(LS-1))^2)^{\frac{1}{2}}$

Determine the total distance along the shock.

199. DTOT = DS(NSKPTS)

The next section will relocate the shock points such that 25% of the points will be along 10% of the total distance along the shock. First determine the number of points in the smaller region, and change to a real number.

200. ND4 = NDJ/4

201. D4 = ND4

202. D4M1 = D4 - 1.0

Determine the smaller incremental distance.

203. DEL = (0.10)(DTOT)/D4M1

Determine the number of points in the larger region, and change to a real number.

204. ND75 = NDJ-ND4

205. D75 = ND75

Determine the larger incremental distance

206. DELP = (0.90)(DTOT)/D75

Now begin relocating the points. The distance starts at the tip of the plug.

207. D = 0.00

208. DO 229 LS = 2,NDJ

Determine the distance where a point is to be set.

209. If LS > ND4, then D = D + DELP

210. If LS $\leq$ ND4, then D = D + DEL

Now set the points using a linear interpolation.

211. XSX(LS) = TAB(D,DS,XS,NSKPTS,1)

212. RSX(LS) = TAB(D,DS,RS,NSKPTS,1)

213. NUSX(LS) = TAB(D,DS,NUS,NSKPTS,1)

214. TSX(LS) = TAB(D,DS,TS,NSKPTS,1)

215. SSX(LS) = TAB(D,DS,SS,NSKPTS,1)

216. SA(LS) = TAB(D,DS,SAS,NSKPTS,1)

217. XSX1(LS) = TAB(D,DS,XS1,NSKPTS,1)

218. RSX1(LS) = TAB(D,DS,RS1,NSKPTS,1)

219. NUSX1(LS) = TAB(D,DS,NS1,NSKPTS,1)

220. TSX1(LS) = TAB(D,DS,TS1,NSKPTS,1)

221. SSX1(LS) = TAB(D,DS,SS1,NSKPTS,1)

Now calculate the downstream variables at each point. First determine the upstream Mach number.

222. CALL PMTURN

Determine the shock wave angle relative to the incoming flow angle.

223. SWAR = SA(LS) - TSX(LS)

Calculate the cotangent of the change in streamline angle across the shock. Then determine this change in angle.

224. COTDEL = TAN(SWAR)((G+1)(M^2)/(2((MS)(SIN(SWAR)))^2-2))-1)

225. DELTA = ATAN(1/COTDEL)

Calculate the downstream streamline angle.

226. TSY(LS) = TSX(LS) + DELTA

Evaluate the downstream variables across the shock at this point.

227. CALL OSHOCK

228. NUSY(LS) = PMANGL(MY,G)

229. SSY(LS) = (DSR)(RG) + SSX(LS)

Reset the total number of shock points.

230. NSKPTS = LS

The characteristics solution will now be restarted. The starting line is along the shock and will be a left-running characteristic from the tip of the plug. First, reset the characteristics matrix to zero.

231. DO237 I = 1,NDI

232. DO237 J = 1,NDJ

233. X(I,J) = 0.0

234. R(I,J) = 0.0

235. NU(I,J) = 0.0

236. T(I,J) = 0.0

237. S(I,J) = 0.0

Determine a real number used in setting the J-subscript.

238. EKJ = 1.60

Begin moving along the characteristic.

239. DO246 I = 2,NSKPTS

Set the J-subscript.

240. EKJ = EKJ + 0.50

241. J = EKJ

Set the points.

242. X(I,J) = XSX(I)

243. R(I,J) = RSX(I)

244. NU(I,J) = NUSY(I)

245. T(I,J) = TSY(I)

246. S(I,J) = SSY(I)

The initial characteristic line has now been set. Now begin the method of characteristics solution which will move from the shock to the near wake surface, and then proceed along a left-running wave until it intersects the shock.

Set the number of characteristic-shock intersections.

247. NINTS = 0

Move from the shock surface to the near wake.

248. DO 384 L = 2,NSKPTS

Determine the streamline angle of the near wake.

249. TW = T2-THET12

Calculate the value of the I-subscript on the near wake surface.

250. IS = (2)(L) - 1

Check for a constant pressure near wake boundary solution.

251. If IS > 3 and IBOUND = 2, then GO TO 254

A conetail near wake boundary is indicated. Calculate the point on the surface.

252. CALL SURF

Now begin moving up the left-running characteristic.

253. GO TO 255

A constant pressure boundary near wake surface is indicated. Calculate the point on the surface.

254. CALL CPB

Increment the starting value of the I-subscript

255. IS1 = IS + 1

Set a real number used in setting the J-subscript

256. EKJ = 1.60

Start moving up the characteristic until an intersection with the shock wave.

257. DO 383 I = IS1,NDIM1

Set the J-subscript

258. EKJ = EKJ + 0.50

259. J = EKJ

See if I is odd or even.

260. MI = MOD(I,2)

261. If MI = 0, then GO TO 264

I is an odd number. Calculate the characteristic point.

262. CALL CALC

No intersection of the characteristic and shock has been found.

263. GO TO 383

I is an even number. Calculate the characteristic point.

264. CALL CALC

No shock-characteristic intersection was found.

265. GO TO 383

The characteristic has intersected the shock. Increment the number of intersections.

266. NINTS = NINTS + 1

See whether I is odd or even.

267. If MI = 0, then GO TO 276

Check for an incomplete characteristics matrix. I is odd.

268. If /X(I-1,J+1)/ < 10^{-6}, then GO TO 385

Determine the subscript of the shock point.

269. JS = I-2-(NINTS-1)

Determine the Mach number and Mach angle at the characteristic point whose characteristic intersected the shock wave.

270. CALL PMTURN

271. EMU2 = ASIN(1./EM2)

Determine the angle of the characteristic, and redefine the coordinates of that characteristic points.

272. ANGCH = T(I-1,J) + EMU2

273. XQ = X(I-1,J)

274. RQ = R(I-1,J)

Continue the calculations in another part of the program.

275. GO TO 283

I is an even number. Check for an incomplete characteristics matrix.

276. If /X(I-1,J) < 10^{-6}, then GO TO 385

Determine the subscript of the shock point.

277. JS = I-2 - (NINTS-1)

Determine the Mach number and Mach angle at the characteristic point whose characteristic intersected the shock wave.

278. CALL PMTURN

279. EMU2 = ASIN(1./EM2)

Determine the angle of the characteristic, and redefine the coordinates of that characteristic point.

280. ANGCH = T(I-1,J-1) + EMU2

281. XQ = X(I-1,J-1)

282. RQ = R(I-1,J-1)

Calculate the slope of the shock and of the characteristic.

283. SLSK = TAN(SA(JS))

284. SLCH = TAN(ANGCH)

Calculate the axial and radial coordinates of the intersection point of the shock and characteristic.

285. XINT = (RQ-RSX(JS) + (SLSK)(XSX(JS)) - (SLCH)(XQ)/(SLSK-SLCH)

286. RINT = RSX(JS) + (SLSK)(XINT-XSX(JS))

Set up a ratio of distances in order to calculate the remaining variables.

287. RATIO = (XINT-XSX(JS))/(XSX(JS+1) - XSX(JS))

Calculate the remaining variables.

288. TINT = TSY(JS) + (RATIO)(TSY(JS+1) - TSY(JS))

289. NUINT = NUSY(JS) + (RATIO)(NUSY(JS+1) - NUSY(JS))

290. SINT = SSY(JS) + (RATIO)(SSY(JS+1) - SSY(JS))

291. NUX = NUSX(JS) + (RATIO)(NUSX(JS+1) - NUSX(JS))

292. TX = TSX(JS) + (RATIO)(TSX(JS+1) - TSX(JS))

293. SX = SSX(JS) + (RATIO)(SSX(JS+1) - SSX(JS))

The next section iterates to obtain a new shock wave angle at this intersection. Make a first estimate of the downstream streamline angle.

294. TYG = T2 + THET12

Check for a positive upstream streamline angle, and then modify this initial estimate of the downstream streamline angle.

295. If TX > 0, then TYG = (1.10)(TX) + 1.0/57.2957795

Determine the change in streamline angle through the shock.

296. DEL = TYG - TX

See if this is greater than the minimum change in streamline angle. If not, then modify it.

297. If DEL < DDEL, then TYG = TX+DDEL

Calculate the change in streamline angle through the shock, and then determine the upstream Mach number.

298. DELTA = TYG - TX

299. CALL PMTURN

Begin the iteration process.

300. DO 312 NCOUNT = 1,100

Calculate the shock angle.

301. CALL SHOCK

Determine the J-subscript for I being odd or even.

302. J1 = J

303. If MI = 0, then J1 = J-1

Calculate the downstream shock variables from the method of characteristics solution.

304. CALL SURFSK

Calculate the downstream shock variables from the shock wave solution.

305. CALL OSHOCK

Determine the downstream Mach number, MY, from this solution.

306. CALL PMTURN

See if the iterations have finished.

307. If NCOUNT = 100, then GO TO 313

Calculate the difference between the downstream Mach numbers as determined by these two different methods.

308. DFMY = MY-MINT

Determine the new change in streamline angle through the shock based on the values of the two downstream Mach numbers determined above.

309. DELTA = (DELTA)(MINT/MY)

Evaluate the next estimate of the downstream streamline angle.

310. TYG = DELTA + TX

See if additional iterations are required.

311. If /DFMY/ < 10^{-4}, then GO TO 313

312. CONTINUE

Set the downstream Prandtl-Meyer angle, shock wave angle and downstream entropy at this point.

313. NUY = NUINT

314. SAR = SWAR + TX

315. SY = (DSR)(RG) + SX

The next section modifies the remainder of the shock wave shape. The numbering system must first be altered. Set the starting values of the I- and J-subscripts.

316. IST = I

317. JST = J

Set a real number used in determining the J-subscript.

318. EKJ = J + 0.10

If I is an even number, then modify this number.

319. If MI = 0, then EKJ = J - 0.40

Begin changing the numbering system.

320. DO 330 IL = IST, NDIM1

Set the J-subscript.

321. EKJ = EKJ + 0.50

322. J = EKJ

See if I is odd or even, and depending on this result, determine the J-subscript of the previous point.

323. MI = MOD(I,2)

324. J1 = J + 1

325. If MI = 0, J1 = J

Reset the characteristic points.

326. X(IL,J) = X(IL-1, J1)

327. R(IL,J) = R(IL-1, J1)

328. NU(IL,J) = NU(IL-1, J1)

329. T(IL,J) = T(IL-1, J1)

330. S(IL,J) = S(IL-1,J1)

The remainder of the shock shape. First establish the intersection as a shock point. Set the starting value of the J-subscript.

331. JS1 = JS + 1

332. XSX(JS) = XINT

333. RSX(JS) = RINT

334. NUSX(JS) = NUX

335. TSX(JS) = TX

336. SSX(JS) = SX

337. NUSY(JS) = NUY

338. TSY(JS) = TY

339. SSY(JS) = SY

340. SA(JS) = SAR

Begin to modify the remainder of the shock shape.

341. DO 368 K = JS1, NDIM1

Calculate the slope of the shock wave.

342. SLSK = TAN(SA(K-1))

Check for an incomplete array of shock points.

343. If XSX(K) < 10^{-6}, then GO TO 368

Calculate the slope of the shock point and the first upstream point.

344. SLCH = (RSX1(K) - RSX(K))/(XSX1(K) - XSX(K))

Determine the intersection of this line and the projected shock.

345. XINT = (RSX(K)-RSX(K-1) + (SLSK)(XSX(K-1)) - (SLCH)(XSX(K)))/
(SLSK-SLCH)

346. RINT = (SLCH)(XINT-XSX(K)) + RSX(K)

Calculate a ratio of distances.

347. RATIO = (XINT-XSX1(K))/(XSX(K)-XSX1(K))

Reset the variables on the upstream side of the shock.

348. XSX(K) = XINT

349. RSX(K) = RINT

350. NUSX(K) = NUSX1(K) + (RATIO)(NUSX(K) - NUSX1(K))

351. TSX(K) = TSX1(K) + (RATIO)(TSX(K) - TSX1(K))

352. SSX(K) = SSX1(K) + (RATIO)SSX(K) - SSX1(K))

Set the first estimate of the downstream streamline angle.

353. TSY(K) = TSY(K-1)

Calculate the change in streamline angle across the shock.

354. DEL = TSY(K) - TSX(K)

See if this change in streamline angle is too small.

355. If DEL < DDEL, then TSY(K) = TSX(K) + DDEL

Calculate the upstream Mach number.

356. CALL PMTURN

Calculate the change in streamline angle across the shock.

357. DELTA = TSY(K) - TSX(K)

Determine the shock wave angle and also the downstream variables.

358. CALL SHOCK

359. CALL OSHOCK

Check for a too low downstream Mach number.

360. If MY < 1.10, then GO TO 362

Everything is satisfactory; go to another part to continue calculations.

361. GO TO 365

Check to see if everything might still be satisfactory.

362. If MINT < 1.10 and MY > 1.01, then GO TO 365

The downstream Mach number is too low. Reset the downstream streamline angle and repeat the calculations.

363. TSY(K) = TSY(K) - 0.0010

364. GO TO 354

Set the remaining downstream variables.

365. NUSY(K) = PMANGL(MY,G)

366. SSY(K) = (DSR)(RG) + SSX(K)

367. SA(K) = SWAR + TSX(K)

368. CONTINUE

The characteristic points must be renumbered to the "lost" point which occurred because of the intersection of the characteristic and the shock. First, see whether the starting value of I is odd or even.

369. MI = MOD(IST,2)

Determine the value of the real number used in setting the J-subscript.

370. EKJ = JST + 0.10

371. If MI = 0, then EKJ = JST - 0.40

Redefine the subscript along the shock.

372. K = JS1

Begin resetting the characteristic points.

373. DO381 K1 = IST, NDIM1

Determine the J-subscript.

374. EKJ = EKJ + 0.50

375. J = EKJ

Now set the characteristic points.

376. X(K1,J) = XSX(K)

477. R(K1,J) = RSX(K)

378. NU(K1,J) = NUSY(K)

379. T(K1,J) = TSY(K)

380. S(K1,J) = SSY(K)

Increment the value of K

381. K = K + 1

The intersection has been found, continue with the solution.

382. GO TO 384

383. CONTINUE

384. CONTINUE

All shock points between the free jet boundary and the tip of the plug have been set. Now set the point on the free jet boundary. First, set the shock wave angle.

385. SA(NSKPTS) = SA(NSKPTS-1)

Determine this shock wave angle relative to the upstream streamline angle.

386. SWAR = SA(NSKPTS) - TSX(NSKPTS)

Calculate the Mach number upstream of the shock on the free jet boundary, and determine its square.

387. CALL PMTURN

388. SM = $MINT^2$

Calculate the change in streamline angle across the shock.

389. COTDEL = TAN(SWAR)(((G+1)(SM)/(2((MINT)(SIN(SWAR))^2-2))-1)

390. DELTA = ATAN(1/COTDEL)

Calculate the downstream streamline angle.

391. TSY(NSKPTS) = TSX(NSKPTS) + DELTA

Determine the variables downstream of the shock.

392. CALL OSHOCK

393. SSY(NSKPTS) = (DSR)(RG) + SSX(NSKPTS)

Evaluate the downstream stagnation pressure, and determine a pressure ratio.

394. POY = (POR)(PO1)

395. PR = POY/PA

Calculate a Mach number and Prandtl-Meyer angle corresponding to this pressure ratio.

396. AMBM = $\left(2\left(\left(PR^{\frac{G-1}{G}}\right)-1\right)/(G-1)\right)^{\frac{1}{2}}$

397. NUAMB = PMANGL(AMBM,G)

Determine the downstream Prandtl-Meyer angle as calculated from the shock solution.

398. CPNUY = PMANGL(MY,G)

Determine the difference in these Prandtl-Meyer angles and then reset the downstream streamline angle.

399. DNU = NUAMB - CPNUY

400. TSY(NSKPTS) = TSY(NSKPTS) + DNU

Set the Prandtl-Meyer angle downstream of the shock.

401. NUSY(NSKPTS) = NUAMB

The next section evaluates the properties across the shock at the intersection of streamlines with the shock. First check to see whether any streamlines are present.

402. If NOSTRL = 0, then GO TO 455

Work with the streamlines not on a boundary.

403. NSL1 = NOSTRL - 1

404. DO 433 JK = 2,NSL1

Determine the final point on each streamline.

405. NSTT = NSTRT(JK)-1

Begin moving along each streamline looking for its intersection with the shock.

406. DO 432 IK = 1,NSTT

Determine the axial coordinate corresponding to an intersection

407. XCALC = TAB(RSTR(JK,IK), RSX,XSX,NSKPTS,1)

Check for an actual intersection.

408. If XSTR(JK,IK) < XCALC, then GO TO 432

An intersection has been found. Calculate the slope of the streamline at the point just upstream of the intersection.

409. SLST = TAN(TSTR(JK,IK)

Determine the shock angle and its slope at the intersection.

410. SACALC = TAB(RSTR(JK,IK), RSX,SA,NSKPTS,1)

411. SLSK = TAN(SACALC)

Determine the coordinates of the actual intersection.

412. XINT = ((SLSK)(XCALC)-(SLST)(XSTR(JK,IK)))/(SLSK-SLST)

413. RINT = (SLSK)(XINT-XCALC) + RSTR(JK,IK)

Now reset the streamline point at the intersection.

414. XSTR(JK,IK) = XINT

415. RSTR(JK,IK) = RINT

Set the remainder of the streamline variables at this point.

416. TSTR(JK,IK) = TAB(RINT,RSX,TSX,NSKPTS,2)

417. NUINT = TAB(RINT,RSX,NUSX,NSKPTS,2)

418. CALL PMTURN

The remaining points on the streamline must be erased.
Set the starting point.

419 IP1 = IK + 1

Begin erasing points.

420. DO 424 KK = IP1,NOSPTS

421. XSTR(JK,KK) = 0.0

422. RSTR(JK,KK) = 0.0

423. MSTR(JK,KK) = 0.0

424. TSTR(JK,KK) = 0.0

The next streamline point will be at the same location, but on the downstream side of the "lip shock".

425. XSTR(JK,IK+1) = XINT

426. RSTR(JK,IK+1) = RINT

Determine the remaining variables at this point.

427. TSTR(JK,IK+1) = TAB(RINT,RSX,TSY,NSKPTS,2)

428. NUINT = TAB(RINT,RSX,NUSY,NSKPTS,2)

429. CALL PMTURN

Reset the starting point on that streamline, and then go to the next streamline.

430. NSTRT(JK) = IK+2

431. GO TO 433

432. CONTINUE

433. CONTINUE

The next section takes care of the streamline along the free jet boundary

434. DO 435 LTP = 1,NOSPTS

Look for the intersection of this streamline with the shock.

435. If XSTR(NOSTRL,LTP) > XSX(NSKPTS) and RSTR(NOSTRL,LTP) > 0.001, then GO TO 436

An intersection has not been found.

436. GO TO 455

An intersection has been indicated. Set the value of the subscript.

437. IINT = LTP

Calculate the slope of the shock and the slope of the boundary.

438. SLSK = TAN(SA(NSKPTS))

439. SLB = TAN(TSTR(NOSTRL,IINT))

Redefine the coordinates of the streamline point.

440. XB = XSTR(NOSTRL,IINT)

441. RB = RSTR(NOSTRL,IINT)

Determine the coordinates of the intersection.

442. XINT = (RB-RSX(NSKPTS) + (SLSK)(XSX(NSKPTS))-(SLB)(XB))/(SLSK-SLB)

443. RINT = (SLB)(XINT-XB) + RB

Reset the last streamline point.

444. XSTR(NOSTRL,IINT) = XINT

445. RSTR(NOSTRL,IINT) = RINT

Calculate the change in streamline angle across the shock.

446. DELTA = TSY(NSKPTS) - TSX(NSKPTS)

Reset the streamline variables upstream of the shock.

447. MSTR(NOSTRL,IINT) = MAMB

448. TSTR(NOSTRL,IINT) = TSX(NSKPTS)

Set the subscript for the next point.

449. IP1 = IINT + 1

Set the streamline point downstream of the shock.

450. XSTR(NOSTRL,IP1) = XINT

451. RSTR(NOSTRL,IP1) = RINT

452. TSTR(NOSTRL,IP1) = TSY(NSKPTS)

453. MSTR(NOSTRL,IP1) = AMBM

Reset the starting point of this streamline.

454. NSTRT(NOSTRL) = IP1 + 1

Redefine the stagnation pressure on this streamline downstream of the shock, and its corresponding Mach number.

455. POLP = POY

456. MAMB = AMBM

Place a characteristic point on the free jet boundary. See whether I is odd or even, and then set the J-subscript.

457. MI = MOD(I,2)

458. JX = J

459. If MI = 0, then JX = J-1

Calculate the free jet boundary point.

460. CALL CPB

Set the integer denoting that recompression has not as yet been reached.

461. IRECP = 0

Set the last values of the I- and J-subscripts.

462. ILAST = I-2

463. JLAST = J-1

Locate the end point of the characteristics matrix.

464. NDF = I-1

Set the starting value of the I-subscript.

465. IST = IS+1

Set the value of the J-subscript.

466. JTOP = J

Reset the last upper characteristic using the average of the characteristic point and the downstream shock point.

467. X(NDF,J) = (X(I,J) + XSX(NSKPTS))/2.0

468. R(NDF,J) = (R(I,J) + RSX(NSKPTS))/2.0

469. NU(NDF,J) = NU(I,J)

470. T(NDF,J) = (T(I,J) + TSY(NSKPTS))/2.0

471. S(NDF,J) = S(I,J)

This next section compeltes the characteristic matrix.

472. DO 510 L = IST,NDF,2

Check for a constant pressure near wake solution.

473. If IBOUND = 2, then GO TO 477

A conetail near wake solution is indicated. Set the near wake streamline angle.

474. TW = T2 - THET12

Calculate the boundary point.

475. CALL SURF

Continue calculations in another section.

476. GO TO 478

A constant pressure boundary is indicated. Calculate the point on this boundary.

477. CALL CPB

See if recompression has been reached.

478. If R(L,1) <RSRB, then IRECP = 1

479. If IRECP = 0, then GO TO 496

Recompression has been reached. Set the value of the I-subscript.

480. IREC = L

Set a ratio of distances and calculate the coordinates and variables at recompression.

481. RATIO = (RSRB-R(L-2,1))/(R(L,1)-R(L-2,1))

483. XREC = X(L-2,1) + (RATIO)(X(L,1) - X(L-2,1))

484. RREC = R(L-2,1) + (RATIO)(R(L,1) - R(L-2,1))

485. V3A = NU(L-2,1) + (RATIO)(NU(L,1) - NU(L-2,1))

486. T3A = T(L-2,1) + (RATIO)(T(L,1) - T(L-2,1))

487. S3A = S(L-2,1) + (RATIO)(S(L,1) - S(L-2,1))

Now reset the characteristic point at recompression.

488. X(L,1) = XREC

489. R(L,1) = RREC

490. NU(L,1) = V3A

491. T(L,1) = T3A

492. S(L,1) = S3A

Determine the Mach number at recompression.

493. CALL PMTURN

Determine the value of the variable SOL3A depending on the type of near wake solution used.

494. If IBOUND = 1, then SOL3A = M3A/M2A

495. If IBOUND = 2, then SOL3A = T3A

Reset the starting value of the I-subscript.

496. IS1 = L+1

Both the "lip shock" and the recompression shock are in the characteristic matrix. The solution will terminate on a left-running characteristic starting at recompression. First set the real number used to determine the J-subscript.

497. EKJ = 1.60

Now continue with the characteristics solution.

498. DO 507 I = IS1,NDJ

Be sure to stop on the left-running characteristic.

499. If IS1 > NDF, then GO TO 510

Set the J-subscript.

500. EKJ = EKJ + 0.50

501. J = EKJ

See whether I is odd or even.

502. MI = MOD(I,2)

503. If MI = 0, then GO TO 506

The subscript I is odd. Calculate the characteristic point and continue calculations in another part of the program.

504. CALL CALC

505. GO TO 507

I is an even number. Calculate the characteristic point.

506. CALL CALC

507. CONTINUE

Calculate streamlines and return to the calling program if recompression has been reached.

508. If IRECP = 1, then CALL STRLNE

509. If IRECP = 1, then RETURN

510. CONTINUE

Recompression has not been reached, so fill in the rest of the characteristic matrix using subroutine FLOW. However, the initial line must be established. First calculate any streamlines.

511. CALL STRLNE

Set integer values.

512. N1 = NSKPTS + 1

513. NOPS = (N1/2) + 1

514. NP1 = NOPS - 1

515. P1 = NP1

The initial line will be along a constant I value line. The term DS is the distance along the shock.

516. DS(1) = 0.0

Redefine the first characteristic point.

517. XSX1(1) = X(NDF,1)

518. RSX1(1) = R(NDF,1)

519. NUSX1(1) = NU(NDF,1)

520. TSX1(1) = T(NDF,1)

521. SSX1(1) = S(NDF,1)

Begin calculating the distance along the line and also redefining these last characteristic points.

522. DO 528 J = 2,NOPS

523. $DS(J) = DS(J-1) + ((X(NDF,J) - X(NDF,J-1))^2 + (R(NDF,J) - R(NDF,J-1))^2)^{\frac{1}{2}}$

524. XSX1(J) = X(NDF,J)

525. RSX1(J) = R(NDF,J)

526. NUSX1(J) = NU(NDF,J)

527. TSX1(J) = T(NDF,J)

528. SSX1(J) = S(NDF,J)

Now set the entire characteristic matrix to zero.

529. DO 535 I = 1,NDI

530. DO 535 J = 1,NDJ

531. X(I,J) = 0.0

532. R(I,J) = 0.0

533. NU(I,J) = 0.0

534. T(I,J) = 0.0

535. S(I,J) = 0.0

Now reset the first point on the initial line.

536. X(1,1) = XSX1(1)

537. R(1,1) = RSX1(1)

538. NU(1,1) = NUSX1(1)

539. T(1,1) = TSX1(1)

540. S(1,1) = SSX1(1)

Set the starting value of the distance along the starting line, and redefine the total distance.

541. D = 0.00

542. SOD = DS(NOPS)

Calculate the incremental distance along the initial line, and then begin moving along this line.

543. DD = SOD/DJM1

544. DO 550 J = 2,NDJ

Determine the distance along the line.

545. D = D + DD

Obtain points on the initial line.

546. X(1,J) = TAB(D,DS,XSX1,NOPS,1)

547. R(1,J) = TAB(D,DS,RSX1,NOPS,1)

548. NU(1,J) = TAB(D,DS,NUSX1,NOPS,1)

549. T(1,J) = TAB(D,DS,TSX1,NOPS,1)

550. S(1,J) = TAB(D,DS,SSX1,NOPS,1)

Reset the final I-subscript.

551. NDF = NDI-1

Redefine the position integer.

552. ILOCN = 2

Continue characteristics solution and then return.

553. CALL FLOW

554. RETURN

Both the "lip shock" and the recompression shock are in the same characteristic matrix. Calculations of the characteristic up to the left-running wave from recompression will now be done. First define the integer.

555. NDJ2 = NDJ/2

Set the starting and ending values of the J-subscript.

556. JST = ((NDF-IREC)/2) + 2

557. JFIN = NDJ2 + 1

Determine the number of I-subscripts between recompression and the end of the characteristics matrix.

558. IDIFF = NDF - IREC

Define three other integers based on this result.

559. ID1 = IDIFF + 1

560. ID2 = IDIFF + 2

561. ID3 = IDIFF + 3

Shift the characteristics matrix to the left.

562. DO 568 I = 1,ID1

563. DO 568 J = 1,JTOP

564. X(I,J) = X(IREC-1+I,J)

565. R(I,J) = R(IREC-1+I,J)

566. NU(I,J) = NU(IREC-1+I,J)

567. T(I,J) = T(IREC-1+I,J)

568. S(I,J) = S(IREC-1+I,J)

Set the last value of the I-subscript.

569. IEND = IDIFF+2

Begin the characteristics calculations.

570. DO 586 K = JST,JTOP

Define an integer.

571. K1 = K+1

Set the real number used to determine the J-subscript.

572. EKJ = K+0.60

Begin moving along right-running waves.

573. DO 584 I = ID2,IEND

Set another integer subscript.

574. IM = I-1

See whether I is odd or even.

575. MI = MOD(I,2)

Determine the J-subscript.

576. EKJ = EKJ - 0.50

577. J = EKJ

See if I is an even number.

578. If MI = 0, then GO TO 582

The subscript I is odd. Calculate the characteristic point and then go to another part to continue. First define a subscript.

579. JP = J + 1

580. CALL CALC

581. GO TO 584

I is even. Define a subscript and then calculate the point.

582. JP = J - 1

583. CALL CALC

584. CONTINUE

Increment the final value of the I-subscript.

585. IEND = IEND + 1

586. CONTINUE

Now move from the top surface of the characteristics matrix to the left-running characteristic.

587. DO 606 K = ID3, NDF, 2

Calculate the J-subscript and then calculate the free jet boundary point.

588. J = JTOP

589. CALL CPB

Define the real number used to determine additional J-subscripts.

590. EKJ = JTOP + 1.60

Set the starting value of the I-subscript.

591. K1 = K + 1

Now begin moving along right-running characteristics.

592. DO 604 I = K1,IEND

Make sure extra points are not calculated.

593. If K1 > IEND, then GO TO 607

Define a calculational integer and the J-subscript.

594. IM = I - 1

595. EKJ - EKJ - 0.50

596. J = EKJ

See if I is odd or even.

597. MI = MOD(I,2)

598. If MI = 0, then GO TO 602

I is odd. Calculate the characteristic point. First define the J-subscript.

599. JP = J + 1

600. CALL CALC

601. GO TO 604

I is even. Set the J-subscript, and calculate the point.

602. JP = J - 1

603. CALL CALC

604. CONTINUE

Increment the final value of the I-subscript.

605. IEND = IEND+1

606. CONTINUE

Redefine the maximum number of J- and I-subscripts.

607. NDJP = NDJ

608. NDIP = NDI

Reset the maximum number of J-and I-subscripts.

609. NDJ = JTOP

610. NDI = (2)(NDJ)

Determine the recompression shock shape.

611. CALL SSHAPE

Reset the maximum number of J- and I-subscripts.

612. NDJ = NDJP

613. NDI = NDIP

614. CONTINUE

The last section takes care of printing out the final variables along the "lip shock". The program skips a page and then prints headings. Now begin moving along the shock.

614. DO 625 K = 1,NSKPTS

Check for a missing point.

615. If XSX(K) < 10^{-6}, then GO TO 625

Dimensionalize the shock point coordinates.

616. XDIM = (XSX(K))(RPB)

617. RDIM = (RSX(K))(RPB)

Change the upstream and downstream streamline angles from radians to degrees.

618. TXD = (TSX(K))(57.2957795)

619. TYD = (TSY(K))(57.2957795)

Determine the upstream and downstream Mach numbers.

620. CALL PMTURN

621. CALL PMTURN

Change the shock wave angle from radians to degrees.

622. SWAD = (SA(K))(57.2957795)

Print out the results at each point.

623. PRINT J,XDIM,RDIM,MINT,MY,TXD,TYD,SWAD,SSX(K),SSY(K)

Punch out the coordinates of the shock point.

624. PUNCH XDIM,RDIM

625. CONTINUE

Return to the calling program.

626. RETURN

627. END

XLIII. MAIN PROGRAM "MAIN"

XLIII. MAIN PROGRAM "MAIN"

The main program reads in all data and non-dimensionalizes distances. After a solution has been reached, the base pressure results and streamline variables are printed out by MAIN.

COMMON BLOCKS

COMMON blocks ACCBLK, AMB, BBBLK, BLBK, BLDM, BLK3A, CH1BLK, CNRANG, CNTR, DATBLK, D3BLK, ETABLK, F4BLK, GAS, PARAM, PBBLK, PTNOS, SIGBLK, SIZE, SOLBLK, STRBLK, STRL, THETBK, TPN, and TRBBLK are used.

TPNZZL SUBROUTINES

MAIN is the main program.

MAIN uses the function subprogram PMANGL, and calls the subroutine TPNZZL.

FORTRAN SYSTEM ROUTINES

Built-in FORTRAN function SQRT is used.

ACC5C6 - accuracy requirement on the base pressure solution

ACST4 - accuracy requirement in locating the j-streamline

CI2 - value of the I_2 integral (Equation 13)

CJ2 - value of the J_2 integral (Equation 15)

CR - ratio of the Crocco number at recompression to that just after separation

CSQ - square of the Crocco number just before separation

C2A - Crocco number just after separation

C3A - Crocco number just upstream of recompression

C3AS - square of C3A

C6 - Crocco number on the d-streamline at recompression (Equation 9)

DELI - boundary layer thickness (inches)

DI - non-dimensional distance along the initial line

DL2ST - boundary layer momentum thickness (inches)

EGP - geometric parameter (left hand side of Equation 1)

ETAD3 - non-dimensional coordinate of the d-streamline at recompression

ETAJ3 - non-dimensional coordinate of the j-streamline at recompression

ETAM2D - non-dimension shift between the viscous and inviscid coordinates just after separation

ETAM3D - non-dimensional shift between the viscous and inviscid coordinates systems at recompression

EX - boundary layer velocity profile exponent

FB - pressure function

G - ratio of specific heats

GD - mass bleed rate (lb_m/sec)

GDBL - equivalent base bleed rate due to a boundary layer (lb_m/sec)

GDRAT - ratio of the base bleed to the sum of the base bleed and the equivalent base bleed due to a boundary layer

GP - geometric parameter (right hand side of Equation 1)

H - non-dimensional bleed number for total bleed

HP - non-dimensional bleed number for base bleed only

I - subscript

IBOUND - integer telling what type of base pressure solution is desired

IOPTR - integer telling the type of initial line

J - subscript

LAM - plug temperature ratio, T_p/T_{o_1}

MAMB - Mach number corresponding to ambient conditions

MBLD - integer denoting whether base bleed and/or a boundary layer is present

MBLDBL - integer denoting whether a boundary layer is present

MESHPM - integer factor used in setting the minimum number of discrete turns from the plug surface to the near wake

MI - Mach number along the initial line

MSTR - array of Mach numbers at each streamline point

M1A - Mach number on the plug surface just before separation

M2A - Mach number just after separation

M3AM2A - ratio of the Mach number just upstream of recompression to M2A

N - denominator of the boundary layer velocity profile exponent

NASHF - recompression coefficient

NDI - maximum number of I-subscripts in the characteristics matrix

NDJ - maximum number of J-subscripts in the characteristics matrix

NEPS - integer telling whether the nozzle configuration is axisymmetric or planar

NEX - denominator of the boundary layer velocity profile exponent

NOIPTS - number of points on the initial line

NOPPTS - number of plug coordinate points

NOSPTS - number of points on each streamline

NOSTRL - number of streamlines

NU - array of Prandtl-Meyer angles (radians) at each point in the characteristic matrix

NUI - array of Prandtl-Meyer angles (radians) at each point along the initial line

OMEGA - conversion factor from degrees to radians

PA - ambient pressure (lb/in^2)

PAF - ambient pressure (lb/ft^2)

PB - base pressure (lb/in^2)

PBPA - ratio of base pressure to ambient pressure

PBPO - base pressure ratio, P_b/P_{o1}

PBP1 - ratio of base pressure to the static pressure just before separation

PEST - initial estimate of the base pressure ratio, P_b/P_{o1}

PHID3 - velocity ratio on the d-streamline at recompression

PHIJ3 - velocity ratio on the j-streamline at recompression

PO1 - chamber stagnation pressure (lb/in^2)

PO1F - chamber stagnation pressure (lb/ft^2)

PRTO1 - static pressure ratio on the plug surface just upstream of separation, P/P_{o1}

R - array of non-dimensional radial coordinates at each point in the characteristics matrix

RDIM - dimensional radial coordinate (used for output)

RG - gas constant ($ft-lb_f/lb_m-{}^oR$)

RI - array of radial coordinates along the initial line

RMACH - Mach number just upstream of recompression

RP - array of radial coordinates at each point on the plug surface

RPB - plug base radius (inches)

RSRB - wake radius ratio

RSTR - array of non-dimensional radial coordinates at each point on a streamline

S - array of entropies at each point in the characteristics matrix ($ft\text{-}lb_f/lb_m\text{-}^{o}R$)

SI - array of entropies along the initial line ($ft\text{-}lb_f/lb_m\text{-}^{o}R$)

SOL3A - equivalent to M3AM2A, OR the streamline angle (radians) just upstream of recompression

S3A - jet spread parameter

T - array of streamline angles (radians) at each point in the characteristics matrix

TDEG - streamline angle in degrees

THET12 - change in streamline angle (radians) from the plug surface to the near wake

THET2A - equivalent to THET12 except that it is in degrees

THET3A - streamline angle (radians) just upstream of recompression

TH3A - equivalent to THET3A except that it is in degrees

TI - array of streamline angles along the initial line (radians)

TID - streamline angle (degrees) at a point along the initial line

TO1 - chamber stagnation temperature (^{o}R)

TO1F - chamber stagnation temperature (^{o}F)

TP - array of streamline angles (radians) at each point along the plug surface

TRB - base temperature ratio, T_b/T_{o_1}

TSTR - array of streamline angles (radians) at each point on a streamline

T2 - streamline angle (radians) on the plug surface at the base

X - array of non-dimensional axial coordinates at each point in the characteristics matrix

XDIM - axial coordinate in inches

XI - array of axial coordinates along the initial line

XP - array of axial coordinates at each point on the plug surface

XPF - non-dimensional axial location of the plug base

XSTR - array of non-dimensional axial coordinates at each point along a streamline

SOLUTION METHOD

Skip a page.

1. PRINT 100

Define the value of 1^{o} in radians

2. OMEGA = 1/57.2957795

Read in the three run-identification cards and print them out immediately.

3. READ 107

4. READ 108

5. READ 109

6. PRINT 107

7. PRINT 108

8. PRINT 109

Read in the number of vertical characteristic points, and calculate the number of horizontal characteristic points.

9. READ 117, NDJ

10. NDI = (2)(NDJ)

Read in the ratio of specific heats, the gas constant, the reservoir temperature and pressure.

11. READ 101, G, RG, TO1, PO1

Read in the base radius and whether the flow is axisymmetric or planar.

12. READ 103, RPB, NEPS

Read in the ambient pressure.

13. READ 106, PA

Print out the ratio of specific heats, the reservoir temperature and pressure and the ambient pressure.

14. PRINT 145, G, TO1, PO1, PA

Read in the number of streamlines to be calculated.

15. READ 146, NOSTRL

Read in the accuracy requirement in locating the j-streamline, the accuracy requirement on the base pressure solution, and the recompression coefficient.

16. READ 110, ACST4, ACC5C6, NASHF

Read in whether a conetail or constant pressure mixing solution is desired.

17. READ 111, IBOUND

Read in whether base bleed is present, and the amount of bleed

18. READ 116, MBLD, GD

Read in whether a finite approaching boundary layer is present, the boundary layer thickness, the profile exponent, and the plug temperature ratio.

19. READ 153, MBLDBL, DELI, N, LAM

Combine the bleed and boundary layer integers.

20. MBLD = MBLD + MBLDBL

Read in the first estimate of the base pressure ratio, and the base temperature ratio.

21. READ 112, PEST, TRB

Read in the number of plug points to be read in, the number of initial points to be read in, and the type of initial line.

22. READ 102, NOPPTS, NOIPTS, IOPTR

Print title and column headings.

23. PRINT 113

24. PRINT 114

Punch out the number of plug points.

25. PUNCH 142, NOPPTS

Read in the plug points, punch out the coordinates, and then print out the coordinates.

26. DO 31 I = 1,NOPPTS

27. READ 104, XP(I), RP(I)

28. PUNCH 143, XP(I), RP(I)

29. PRINT 115, XP(I), RP(I)

Non-dimensionalize coordinates with the base radius.

30. XP(I) = XP(I)/RPB

31. RP(I) = RP(I)/RPB

Redefine the axial distance of the last plug point.

32. XPF = XP(NOPPTS)

Now read in the position and variables along the initial line.

33. DO 38 J = 1,NOIPTS

34. READ 105, XI(J), RI(J), MI,TID, SI(J)

Change the streamline angle from degrees to radians.

35. TI(J) = (TI(J))(OMEGA)

Calculate the Prandtl-Meyer (P-M) angle corresponding to the Mach number at this point.

36. NUI(J) = PMANGL(MI,G)

Non-dimensionalize the coordinates.

37. XI(J) = XI(J)/RPB

38. RI(J) = RI(J)/RPB

Determine the solution to the nozzle base pressure and flow field.

39. CALL TPNZZL

Skip a page.

40. PRINT 100

See if streamlines have been calculated.

41. If NOSTRL = 0, then GO TO 57

Print out streamline locations and variables.

42. DO 55 J = 1, NOSTRL

Skip a page and print title and column headings.

43. PRINT 100

44. PRINT 119

45. PRINT 120, J, NOSTRL

46. PRINT 121

Move along each streamline.

47. DO 55 I = 1,NOSPTS

Look for the last point on the streamline.

48. If RSTR(J,I) $< 10^{-6}$, then GO TO 55

Look for an intersection between the lip shock and a streamline.

49. If I > 1, and /XSTR(J,I)-XSTR(J,I-1)/ $< 10^{-5}$, then PRINT 147

Dimensionalize coordinates, and change the streamline angle from radians to degrees.

50. XDIM = (XSTR(J,I))(RPB)

51. RDIM = (RSTR(J,I))(RPB)

52. TDEG = (TSTR(J,I))(57.2957795)

Print out coordinates and variables and punch out coordinates.

53. PRINT 142, XDIM, RDIM, MSTR(J,I), TDEG

54. PUNCH 143, XDIM, RDIM

55. CONTINUE

Skip a page.

56. PRINT 100

Print base pressure titles.

57. PRINT 123

58. PRINT 124

Print out the ratio of specific heats and the gas constant.

59. PRINT 125, G, RG

Calculate the temperature in degrees F.

60. TO1F = TO1 - 459.0

Print the reservoir temperature in ^{o}R and ^{o}F

61. PRINT 126, TO1, TO1F

Calculate the reservoir pressure and ambient pressure in psfa.

62. PO1F = (PO1)(144)

63. PAF = (PA)(144)

Print out the reservoir and ambient pressures.

64. PRINT 127, PO1, PO1F, PA, PAF

Print out the base temperature ratio.

65. PRINT 128, TRB

Determine the actual nozzle base bleed; subtract off the equivalent base bleed due to the initial boundary layer. Print out this bleed rate.

66. GD = GD - GDBL

67. PRINT 129, GD

Calculate the non-dimensional bleed number.

68. HP = (GDRAT)(H)

If base bleed is present, print out this non-dimensional bleed number.

69. If MBLD = 1, or MBLD = 3, then PRINT 144, HP

See if a boundary layer is present.

70. If MBLD = 1, or MBLD = 0, then GO TO 79.

Print boundary layer title.

71. PRINT 154

Print out the boundary layer and momentum thicknesses.

72. PRINT 152, DELI, DL2ST

Change the profile exponent to an integer number.

73. NEX = N + 0.0010

Print out the profile exponent and the plug temperature ratio.

74. PRINT 149, NEX, LAM

Print out the equivalent bleed due to the boundary layer.

75. PRINT 150, GDBL

Calculate the equivalent non-dimensional bleed number due to the initial boundary layer.

76. HP = (1.0 - GDRAT)(H)

Print out this result.

77. PRINT 151, HP

78. GO TO 80

No boundary layer is present. Print out a message to this effect.

79. PRINT 148

Determine the type of mixing solution used and print out a message.

80. If IBOUND = 1, then PRINT 130

81. If IBOUND = 2, then PRINT 131

Print out recompression coefficient.

82. PRINT 132, NASHF

Print output title.

83. PRINT 133

Calculate the actual base pressure, and determine the base-to-ambient pressure ratio.

84. PB = (PBPO)(PO1)

85. PBPA = PB/PA

Calculate the static pressure ratio just upstream of separation and calculate another base pressure ratio.

86. PRTO1 = $(1 + ((G-1)/2)(M1A^2))^{\frac{G}{G-1}}$

87. PBP1 = (PBPO)(PRTO1)

Print out base pressure ratios.

88. PRINT 134, PBPO, PBPA, PBP1

Calculate the Mach number upstream of recompression.

89. RMACH = (M3AM2A)(M2A)

Print out the Mach numbers.

90. PRINT 135, M1A, M2A, RMACH

Change the wake angles from radians to degrees, print out these results.

91. THET2A = - (THET12)(57.2957795)

92. TH3A = - (THET3A)(57.2957795)

93. PRINT 136, THET2A, TH3A

Print out the jet spread parameter and the wake radius ratio.

94. PRINT 137, S3A, RSRB

Calculate the Crocco numbers and print out the results.

95. C3A = SQRT(C3AS)

96. C2A = C3A/CR

97. PRINT 138, C2A, C3A, C6

Print out conditions along the j- and d-streamlines.

98. If MBLD = 0, then PRINT 139, PHIJ3, ETAJ3

99. If MBLD $\neq$ 0, then PRINT 140, PHIJ3, ETAJ3, PHID3, ETAD3

If the flow is axisymmetric, print out the displacement.

100. If NEP $\neq$ 0, then PRINT 141, GP, ETAM2D, ETAM3D

101. STOP

102. END